

Kosti Herranen

## **OHJELMISTON JAKELU WINDOWS POWERSHELLIN AVULLA**

## **OHJELMISTON JAKELU WINDOWS POWERSHELLIN AVULLA**

Kosti Herranen  
Opinnäytetyö  
Kevät 2014  
Tietotekniikan koulutusohjelma  
Oulun ammattikorkeakoulu

# TIIVISTELMÄ

Oulun ammattikorkeakoulu  
Tietotekniikan koulutusohjelma, ohjelmistojen kehitys

---

Tekijä: Kosti Herranen  
Opinnäytetyön nimi: Ohjelmiston jakelu Windows PowerShellin avulla  
Työn ohjaaja: Lauri Pirttiaho  
Työn valmistumislukukausi ja -vuosi: Kevät 2014  
Sivumäärä: 51 + 0 liitettä

---

Windows XP on ollut yritysmaailmassa käytössä aivan näihin päiviin saakka. Yrityksissä käytetään useita eri ohjelmia ja ratkaisuja, joista osa ei välttämättä ole yhteensopivia uudemman käyttöjärjestelmän kanssa. Tämä on osaltaan hidastanut yrityksiä siirtymistä uuteen käyttöjärjestelmään aikaisemmin. Microsoftin tuki XP-käyttöjärjestelmälle on päättymässä huhtikuussa 2014, minkä johdosta viimeisetkin yritykset ovat aloittaneet Windows 7 -käyttöönottoprojektit kiireellisillä aikatauluilla.

Tämän opinnäytetyön aiheena oli määritellä, suunnitella, toteuttaa ja testata Windows 7 -yhteensopiva ohjelmiston jakelumenetelmä. Työn toimeksiantaja toimittaa palvelin- ja työasemaohjelmista koostuvaa ohjelmistokokonaisuutta asiakkailleen. Asiakasorganisaatioiden ATK-ylläpito asentaa jakelumenetelmän avulla työasemaohjelmistoon kohdistuvia päivityksiä keskitetysti ja etänä. Perinteinen ohjelmiston jakelumenetelmä ei ollut yhteensopiva Windows 7 -käyttöjärjestelmän kanssa, mihin tässä työssä lähdettiin hakemaan ratkaisua.

Ratkaisu päätettiin toteuttaa Windows PowerShellin avulla, joka on tullut osaksi käyttöjärjestelmää Windows 7:stä lähtien. Kehitystyö tehtiin PowerShellin omalla PowerShell ISE -työkalulla. Ohjemateriaalina käytettiin Microsoftin MSDN-sivuston artikkeleita, internetin hakukoneita sekä PowerShellin omaa ohjetietokantaa. Työn lopputuloksena syntyi Windows 7 -yhteensopiva uusi ohjelmiston jakelumenetelmä, jonka avulla ohjelmistopäivitykset voidaan asentaa ympäristön kaikille työasemille keskitetysti ja etänä.

PowerShell osoittautui projektin aikana tehokkaaksi työkaluksi tietokoneiden ylläpidossa ja automatisoinnissa. PowerShell löytyy jatkossa kaikista Microsoftin uusista käyttöjärjestelmistä, joten se tulee ottamaan merkittävän roolin järjestelmien ylläpitotyössä.

---

Asiasanat:  
Windows, PowerShell, ohjelmistojakelu, skriptit, AD, GPO

# SISÄLLYS

|   |    |
|---|----|
| TIIVISTELMÄ                                       | 3  |
| SISÄLLYS  | 4  |
| SANASTO   | 6  |
| 1 JOHDANTO  | 8  |
| 2 VAATIMUSMÄÄRITTELY                              | 9  |
| 2.1 Asiakkaat                                     | 9  |
| 2.2 Käyttäjät                                     | 9  |
| 2.3 Käyttötarkoitus                               | 9  |
| 2.4 Käyttöympäristöt                              | 10 |
| 2.4.1 Palvelimet                                  | 10 |
| 2.4.2 Työasemat                                   | 10 |
| 2.4.3 Verkkoyhteydet                              | 10 |
| 2.5 Asiakasvaatimukset                            | 11 |
| 2.5.1 Käyttötilanteet                             | 11 |
| 2.5.2 Käytettävyys                                | 11 |
| 2.5.3 Tietoturva                                  | 12 |
| 2.6 Käyttäjien ilmoittamia ongelmia               | 12 |
| 3 SUUNNITTELU                                     | 14 |
| 3.1 Toteutustavan valinta                         | 14 |
| 3.1.1 SCCM  | 14 |
| 3.1.2 Perinteinen Windows XP -jakelumenetelmä     | 15 |
| 3.1.3 Windows PowerShell                          | 17 |
| 3.2 Aktiviteettikaavio                            | 20 |
| 3.3 Käyttöliittymät                               | 22 |
| 3.3.1 Palvelimen käyttöliittymä                   | 22 |
| 3.3.2 Työaseman näkymä                            | 25 |
| 3.4 Menetelmän käynnistäminen työasemilla         | 26 |
| 3.4.1 Tehtävien ajoittaminen                      | 27 |
| 3.4.2 Ajoitetun tehtävän levittäminen työasemille | 28 |

|  |    |
|--|----|
| 4 TOTEUTUS   | 29 |
| 4.1 Ratkaistavat ongelmat                                | 29 |
| 4.1.1 Asennuksen vaatimat käyttöoikeudet                 | 29 |
| 4.1.2 Käyttäjälle näytettävät ilmoitukset                | 30 |
| 4.1.3 Jakelumenetelmän käynnistäminen halutulla hetkellä | 33 |
| 4.2 Skriptit   | 34 |
| 4.2.1 Setup.ps1  | 34 |
| 4.2.2 CreateVersionFile.ps1                              | 35 |
| 4.2.3 SoftwareUpdate.cmd                                 | 36 |
| 4.2.4 InitInstall.ps1                                    | 37 |
| 4.2.5 CreateScheduledTask.cmd                            | 37 |
| 4.2.6 UpdateSoftwareClient.ps1                           | 38 |
| 4.3 Palvelin- ja työasema-arkkitehtuuri                  | 41 |
| 5 TESTAUS  | 43 |
| 5.1 Pilotointi   | 43 |
| 5.2 Sisäinen testaus                                     | 44 |
| 5.3 Asiakastestaus                                       | 44 |
| 6 JATKOKEHITYS   | 46 |
| 6.1 Omia havaintoja                                      | 46 |
| 6.2 Asiakaspalautteita                                   | 47 |
| 7 YHTEENVETO   | 48 |
| LÄHTEET  | 50 |

## SANASTO

|                 |   |
|-----------------|---|
| AD              | Active Directory. Microsoft Windows -ympäristöjen käyttäjä- ja laitetietokanta sekä hakemistopalvelu. (1.)  |
| Cmd-tiedosto    | Windows NT -pohjaisen käyttöjärjestelmän komentojotiedosto, jonka avulla suoritetaan komentosarjoja.  |
| GPO             | Group Policy Object. AD-toimialueessa käytetty hallintamekanismi, jonka avulla voidaan luoda ja linkittää asetuksia useille tietokoneille ja käyttäjille samanaikaisesti. (2.)  |
| Järjestelmätili | Local system account. Järjestelmänvalvojan tunnustakin vahvempi paikallinen tunnus Windows-käyttöjärjestelmässä. Tilillä on täydet oikeudet paikalliseen käyttöjärjestelmään, mutta hyvin rajalliset oikeudet verkon resursseihin tai toisiin tietokoneisiin. Järjestelmätili esiintyy verkossa toisille tietokoneille konetilin nimellä. Useat käyttöjärjestelmän prosesseista suoritetaan tämän tilin avulla. |
| OU              | Organizational Unit. AD-objekti, jonka sisälle voidaan sijoittaa muita AD-objekteja, kuten tietokoneita, käyttäjiä, tulostimia tai toisia OU-objekteja hierarkkiseen puurakenteeseen. OU-objektia voidaan kutsua AD:n puurakenteessa myös OU-haaraksi. (3.)   |
| PowerShell      | Windows PowerShell on uusien Windows-käyttöjärjestelmien kehittynyt komentotulkki ja .NET Framework -kirjastoon perustuva komentosarjakieli. Käytetään järjestelmien ylläpitoon ja automatisointiin. On sisäänrakennettuna ominaisuutena Windows 7:ssä  |

ja sitä uudemmissa käyttöjärjestelmissä, mutta on myös asennettavissa erillisellä asennuspaketilla vanhempiin käyttöjärjestelmiin. (4.)

|              |  |
|--------------|--|
| Ps1-tiedosto | PowerShell-komentoja sisältävä komentojonotiedosto.  |
| PsExec       | Järjestelmien ylläpitoon kehitetyn Sysinternals Suite -apuohjelmapaketin yksittäinen sovellus, jonka avulla voidaan suorittaa Windows-käyttöjärjestelmässä komentoja eri käyttäjänä joko paikallisesti tai toisella laitteella. (5.) |
| SCCM         | System Center Configuration Manager. Microsoftin kehittämä työasemien ja palvelinympäristöjen hallintajärjestelmä. (6.)  |
| UAC          | User Account Control eli käyttäjätilien hallinta. Uusien Windows-käyttöjärjestelmien tietoturvaa parantava ominaisuus, joka estää ohjelmia tekemästä muutoksia käyttöjärjestelmään ilman käyttäjän antamaa valtuutusta. (7.)         |
| UNC          | Lyhenne sanoista Uniform Naming Convention eli yhdenmukainen nimeämiskäytäntö. Levyaseman tunnuksen ja polun sijasta UNC-nimessä käytetään syntaksia \\palvelin\jaettu_kansio\polku\tiedosto.  |

# 1 JOHDANTO

Tämän työn aiheena oli kehittää ohjelmiston jakelumenetelmä Windows 7 -käyttöjärjestelmälle. Ohjelmiston toimittaja toimittaa ohjelmistokokonaisuuden, joka muodostuu työasema- ja palvelinohjelmista. Työasemaohjelmien täytyy olla päivitettävissä keskitetysti ja etänä haluttuna ajankohtana, mihin tässä työssä kehitettiin ratkaisu.

Microsoft paransi Windows Vistan myötä merkittävästi Windows-käyttöjärjestelmän tietoturvaa, missä yksi tärkeimmistä uudistuksista oli User Account Control eli UAC. Se estää ohjelmia tekemästä muutoksia käyttöjärjestelmään ilman käyttäjän antamaa valtuutusta. UAC estää samalla työasemien automatisointiin suunniteltujen vanhempien ohjelmien toiminnan, joista yhtenä voidaan mainita ohjelmiston asennuksessa käytetty perinteinen jakelumenetelmä.

Tässä työssä lähdettiin etsimään perinteiselle jakelumenetelmälle modernia vaihtoehtoa. Ensin selvitettiin valmiita vaihtoehtoja ohjelmistojakeluun Windows 7 -käyttöjärjestelmälle. Varteenotettava kaupallinen jakelumenetelmä ei soveltunut kohderyhmälle kalliiden hankinta- ja lisenssikustannusten sekä haastavan ylläpidon vuoksi. Tämän lisäksi selvitettiin mahdollisuutta muokata perinteinen jakelumenetelmä Windows 7 -yhteensopivaksi. Esteeksi muodostui tietoturvan heikkeneminen käyttöönoton kustannuksella.

Koska soveltuvaa valmista tuotetta ei löydetty, päätettiin uusi jakelumenetelmä kehittää IT-ammattilaisten suosimalla PowerShell -työkalulla. Työ piti sisällään määrittely-, suunnittelu-, toteutus- ja testausvaiheet.



## **2 VAATIMUSMÄÄRITTELY**

### **2.1 Asiakkaat**

Jakelumenetelmän kohderyhmään kuuluvat kokoluokaltaan pienet (maksimi 100 käyttäjää) sekä keskisuuret (100–500 käyttäjää) asiakasorganisaatiot. Kohderyhmässä ovat myös ne suuret organisaatiot (yli 500 käyttäjää), joilla joko ei ole käytettävissä valmiita kaupallisia jakelumenetelmiä tai käytössä oleva ratkaisu ei sovellu riittävän hyvin ohjelmiston päivittämiseen.

### **2.2 Käyttäjät**

Toteutuksen avulla asiakasorganisaatioiden ATK-tukihenkilöt, ohjelmiston pääkäyttäjät tai ohjelmiston toimittajan asiantuntijat päivittävät työasemaohjelmistoa. Käyttäjien tietotekniikan tuntemus vaihtelee vähäisestä erittäin kokeneeseen. Loppukäyttäjät ovat asiakasorganisaatioihin kuuluvia henkilöitä, jotka käyttävät päivitettävää ohjelmistoa työasemillaan. Asiakasorganisaation tukihenkilöillä on yleensä järjestelmänvalvojan oikeudet päivitettävälle työasemille, mutta loppukäyttäjillä on tavalliset käyttöoikeudet.

### **2.3 Käyttötarkoitus**

Kehitettävä menetelmä tulee korvaamaan asiakkaiden käyttämän perinteisen Windows XP -jakelumenetelmän, joka ei ole yhteensopiva Windows 7 -työasemien päivittämiseen. Toteutuksen avulla ohjelmiston asennuksista vastaavat henkilöt asentavat työasemaohjelmia loppukäyttäjien työasemille. Normaali käytötapaus on päivittää työasemaohjelmisto etänä keskitetysti uusimpaan versioon kaikille ympäristön työasemille samanaikaisesti. Päivitys tehdään palvelinohjelmiston päivityksen yhteydessä. Jakelumenetelmän merkittävin tavoiteltava hyöty on päivittämiseen käytetyn ajan vähentäminen ja päivitysten toimitusvarmuuden parantaminen.

## **2.4 Käyttöympäristöt**

### **2.4.1 Palvelimet**

Jakelumenetelmä sijoitetaan asiakkaan omistamille käytössä oleville palvelimille, joten käyttöönotosta ei tule laitehankintoja. Palvelimilla suoritetaan ohjelmiston palveluita ja resursseja, joiden toimintaa jakelumenetelmä ei saa häiritä.

Alustapalvelin on joko fyysinen tai virtuaalinen Windows Server 2003, Windows Server 2003 R2, Windows Server 2008 tai Windows Server 2008 R2. Palvelinlaitteiston ikä on korkeintaan kolme vuotta. Tämä täyttää ohjelmiston asettamat laitevaatimukset.

### **2.4.2 Työasemat**

Päivitysten pääasiallisena käyttökohteena ovat Windows 7 -työasemat. Menetelmän tulee olla yhteensopiva myös Windows XP -työasemien päivittämiseen, sillä XP on asiakkailla vielä yleisesti käytössä ja usean rinnakkaisen jakelutavan ylläpitäminen voi olla hankalaa. Osa asiakkaista tulee kuitenkin päivittämään XP-työasemat perinteisellä jakelumenetelmällä, kunnes työasemat on ehditty korvata uusilla. Microsoftin tuki XP:lle päättyy 8.4.2014, minkä jälkeen XP-työasemien lukumäärän tiedetään pienentyvän merkittävästi ja poistuvan lopulta täysin ympäristöistä. Päivitettävien työasemien määrä on suoraan verrannollinen asiakasympäristön kokoluokkaan.

### **2.4.3 Verkkoyhteydet**

Työasemat ja palvelimet liikennöivät yleensä samassa lähiverkossa, mutta joissakin tapauksissa osa työasemista saattaa sijaita etätyöpisteessä hitaiden DSL-tekniikalla toteutettujen verkkoyhteyksien päässä. Työasemat ja palvelimet saattavat sijaita eri lähiverkoissa, jotka ovat yhdistettynä WAN-verkkotekniikan avulla. Työasemien verkkoliitännän nopeus on vähintään 10 Mbit/s ja palvelinten 100–1000 Mbit/s. Verkkojen nopeudet ja toteutukset vaihtelevat asiakkaiden IT-infrastruktuurin mukaan.

## **2.5 Asiakasvaatimukset**

### **2.5.1 Käyttötilanteet**

Päivitettävä ohjelmisto on luokiteltu saatavuudeltaan kriittiseksi, joten päivitykset tehdään yleensä normaalin toimistotyöajan (klo 7.00–17.00) ulkopuolella. Ohjelmiston toimittaja päivittää ensin palvelinohjelmiston uuteen versioon. Työasemaohjelmien uudet asennuspaketit siirretään sovittuun verkkoresurssiin joko ennen palvelinohjelmiston päivitystä tai sen jälkeen. Ohjelmiston pääkäyttäjä tai ATK-osaston henkilö asentaa työasemille uudet työasemaohjelmat etänä yhdestä sijainnista. Loppukäyttäjät kirjautuvat aamulla työasemalle ja alkavat käyttämään päivitettyä ohjelmistoa.

Loppukäyttäjät sulkevat usein työasemat työpäivän päätteeksi. Päivityksen aikana suljetut työasemat on päivitettävä seuraavana aamuna ennen ohjelmiston käyttöä.

Työasemien määrä asiakasorganisaatioissa vaihtelee muutamasta useisiin satoihin, ja ohjelmistoa päivittävät henkilöt valvovat ohjelmistojakelun etenemistä näillä työasemilla yhdestä sijainnista etänä.

### **2.5.2 Käytettävyys**

Ohjelmistopäivityksen tulee olla helposti ja nopeasti jaeltavissa kaikille työasemille samanaikaisesti. Jakelun etenemisen tulee olla reaaliaikaisesti seurattavissa yhdestä sijainnista. Jakelumenetelmän palvelinkäyttöliittymän ei tarvitse olla graafinen, mutta sen käytön tulee olla mahdollisimman yksinkertaista ja suoraviivaista. Käsien kirjoitettavien asetuksien määrän tulee olla mahdollisimman vähäinen inhimillisten virheiden välttämiseksi.

Ohjelmisto sisältää useita asennuspaketteja, joiden lukumäärä riippuu asiakkaan ohjelmistoon tilaamista ominaisuuksista. Pakettien asennuksessa tulee noudattaa ohjelmiston käyttöohjeissa ilmoitettua asennusjärjestystä. Jokaisella ohjelmistoon asennettavalla ominaisuudella on oma versionumeronsa. Ominaisuuden version tulee olla yhteensopiva palvelinohjelmiston kanssa. Epäyhteen-

sopivan ohjelman käyttö on estettävä työasemalla automaattisesti, jos ohjelman versio ei vastaa palvelimen tukemaa versiota.

Ohjelmistojakelun tulee olla pysäytettävissä kaikilla työasemilla keskitetysti ja nopeasti, jotta ongelmatilanteen ilmetessä virheellisen päivityksen jakelu työasemille voidaan keskeyttää.

Loppukäyttäjille ratkaisun tulee olla mahdollisimman huomaamaton, eikä heiltä vaadita mitään syötettä päivityksen alkaessa tai edetessä. Loppukäyttäjää on kuitenkin hyvä informoida päivityksen etenemisestä ja lopputuloksesta.

### **2.5.3 Tietoturva**

Palvelimella jakelumenetelmän asetuksien muokkaamisen tulee olla rajattuna hyvin pienelle käyttäjäryhmälle, johon kuuluvat asiakkaan ATK-tukihenkilöt, ohjelmiston pääkäyttäjät sekä ohjelmiston toimittaja. Loppukäyttäjillä tulee olla luku-oikeudet palvelimella sijaitseviin tiedostoihin, joten palvelimelle ei saa sijoittaa arkaluontoista tietoa, kuten järjestelmänvalvojan käyttäjätunnuksia. Jakelumenetelmällä tulee olla riittävät oikeudet asentaa ohjelmia työasemille, vaikka loppukäyttäjällä olisi tavalliset käyttöoikeudet omalle työasemalle.

## **2.6 Käyttäjien ilmoittamia ongelmia**

Asiakkaat ovat ilmoittaneet, että perinteisellä jakelumenetelmällä työasemille kirjautuminen on toisinaan hyvin hidasta, vaikka päivitettävä ohjelmisto olisi valmiiksi ajan tasalla. Ongelman aiheuttaa jakelumenetelmän tapa tarkistaa päivitettävän ohjelmiston ajantasaisuus.

Päivityksen etenemisestä ja onnistumisesta ei myöskään tule selkeää informaatiota loppukäyttäjille. Tästä aiheutuu ylimääräistä vaivaa ja hämmennystä loppukäyttäjien käynnistäessä usein ohjelmiston liian aikaisin, mikä voi johtaa päivityksen epäonnistumiseen. Tästä seuranneita ongelmia ylläpitäjät joutuvat korjaamaan käsin asentamalla ohjelmia uudelleen työasemille.

Perinteisen jakelumenetelmän ylläpito vaatii myös paljon asetuksien käsin kirjoittamista, mikä on usein johtanut inhimillisiin virheisiin ja päivityksen epäonnistumiseen.

### **3 SUUNNITTELU**

Suunnitteluvaiheen alussa käytin reilusti aikaa selvittäen eri vaihtoehtoja Windows 7 -yhteensopivan jakelumenetelmän löytämiseen. Toteutustavan selvittyä rakensin itselleni kehitysympäristön, joka koostui virtuaalisista Windows 7- ja Windows XP -työasemista Active Directory -toimialueella. Kokeilin aktiivisesti eri vaihtoehtoja testiympäristössä jo tässä vaiheessa ja tallensin hyväksi havaitsemani ratkaisut varsinaista toteutusta varten.

#### **3.1 Toteutustavan valinta**

Windows 7 -työasemien jakelumenetelmän valinnassa selvitettiin toteutuksen kokonaishintaa, käytettävyyden helppoutta sekä soveltuvuutta käyttötarkoitukseen. Kaupallisista vaihtoehdoista tarkempaan vertailuun valittiin SCCM, ja myös perinteisen jakelumenetelmän sovittamista Windows 7:lle selvitettiin.

##### **3.1.1 SCCM**

###### **Taustaa**

Vuonna 1994 Microsoft julkaisi tuotteesta ensimmäisen version nimellä Systems Management Server, eli SMS. Järjestelmä kantaa nykyisin nimeä SCCM, joka tunnetaan edistyneenä työkaluna suurten työasema- ja palvelinympäristöjen hallintaan.

###### **Ominaisuudet**

SCCM:n tärkeimpiä ominaisuuksia ovat laiteasetuksien täydellinen hallinta, etäyhteydet, päivitysten hallinta, ohjelmistojen jakelu, käyttöjärjestelmien jakelu, verkon käytön suojaaminen sekä laite- ja sovellusinventarioiden luonti. (8, linkki Introduction to Configuration Manager.)

SCCM skaalautuu erinomaisesti kaiken kokoisiin ympäristöihin. Tuotteen avulla on mahdollista hallita maksimissaan jopa 400 000 työasemaa keskitetysti sekä melkein yhtä suurta määrää mobiililaitteita. (9, linkki Supported Configurations for Configuration Manager.)

## **Ylläpito**

Monipuolisuutensa vuoksi SCCM:n käyttöönoton suunnittelu ja asentaminen vaatii ylläpidolta koulutuksien kautta hankittua ammattitaitoa. Osaavalle IT-asiantuntijalle SCCM on erinomainen työkalu koko työasema- ja palvelininfran tehokkaaseen hallintaan, mutta vähemmän kokeneelle ylläpitäjälle työkalun käyttäminen voi olla erittäin hankalaa.

## **Soveltuvuus**

SCCM on luultavasti liian monimutkainen työkalu pieniin ympäristöihin sekä moniin keskisuuriin ympäristöihin. Tuotteen käyttöönotto on luultavasti mahdollista vasta asiakkaan IT-asiantuntijan kouluttautumisen jälkeen. SCCM vaatii koulutuksen lisäksi palvelin- ja lisenssihankintoja sekä huolellista suunnittelua asiakkaan ympäristössä. Ylimääräiset kulut ovat juuri niitä, joita kohderyhmän asiakkaat haluavat välttää. Kokonaiskustannuksiltaan SCCM ei sovellu tarkoitukseen, mutta teknisiltä ominaisuuksiltaan tuote täyttää erinomaisesti kaikki ohjelmiston jakeluun liittyvät vaatimukset.

### **3.1.2 Perinteinen Windows XP -jakelumenetelmä**

#### **Taustaa**

Perinteinen jakelumenetelmä on ohjelmiston toimittajan 2000-luvun alkupuolella kehittämä ratkaisu työasemaohjelmien päivittämiseen. Se on kohderyhmässä tähän päivään saakka yleisessä käytössä ollut ratkaisu, jonka avulla ohjelmiston päivityksiä on jaeltu XP-työasemille.

Menetelmä käynnistää määritetystä sijainnista kaikki ohjelmiston asennuspaketit yksitellen ja vertaa paketin versionumeroa työasemalle asennettuun ohjelmaan. Asennus suoritetaan vain, jos asennettu versio on asennuspaketin versiota vanhempi.

Toteutuksessa on hyödynnetty lisäksi kolmannen osapuolen valmistamaa apuohjelmaa, joka mahdollistaa asennukset järjestelmävalvojan oikeuksilla, vaikka kirjautuneella käyttäjällä olisi tavalliset käyttäjätason oikeudet työasemalle. Rat-

kaisu toimii ainoastaan Windows 2000 -, Windows XP - ja Windows Server 2003 -käyttöjärjestelmillä.

Windows Vista ei koskaan yleistynyt yritysmaailmassa. Sen sijaan yrityksissä käytettiin pitkään Windows XP:tä ja siitä siirrytään suoraan Windows 7 - käyttöjärjestelmään. Windows 7:ssä suurin yksittäinen este perinteisen toteutuksen käyttämiselle on UAC, joka estää tehokkaasti haittaohjelmien muutokset työasemalla, mutta estää samalla perinteisen jakelumenetelmien käytön. (11.)

### **Ominaisuudet**

Perinteinen jakelumenetelmä mahdollistaa ohjelmiston päivittämisen työasemalle, vaikka koneelle kirjautuneella käyttäjällä ei olisi järjestelmänvalvojan oikeuksia. Menetelmää kutsutaan Windowsin kirjautumisrutiinien yhteydessä, joten menetelmä skaalautuu alustan ja verkon asettamissa rajoissa suurellekin työasemamäärälle. Menetelmä kirjoittaa työasemakohtaista lokia asetuksissa määritettyyn sijaintiin.

### **Ylläpito**

Ratkaisu on ollut jo pitkään käytössä, joten toteutus ymmärretään hyvin asiakaskunnassa. Jakelumenetelmän määrittämisestä ohjelmistopäivityksessä on vastannut yleensä ohjelmiston toimittaja, mutta joissakin tapauksissa myös asiakkaan IT-asiantuntija tai ohjelmiston nimetty pääkäyttäjä. Muutoksien tekeminen on kohtalaisen helppoa, mutta vaatii paljon käsin kirjoittamista, mikä johtaa usein kirjoitusvirheisiin. Konfiguraation virheet saattavat näkyä ohjelmistopäivitysten heikentyneenä toimitusvarmuutena.

### **Soveltuvuus**

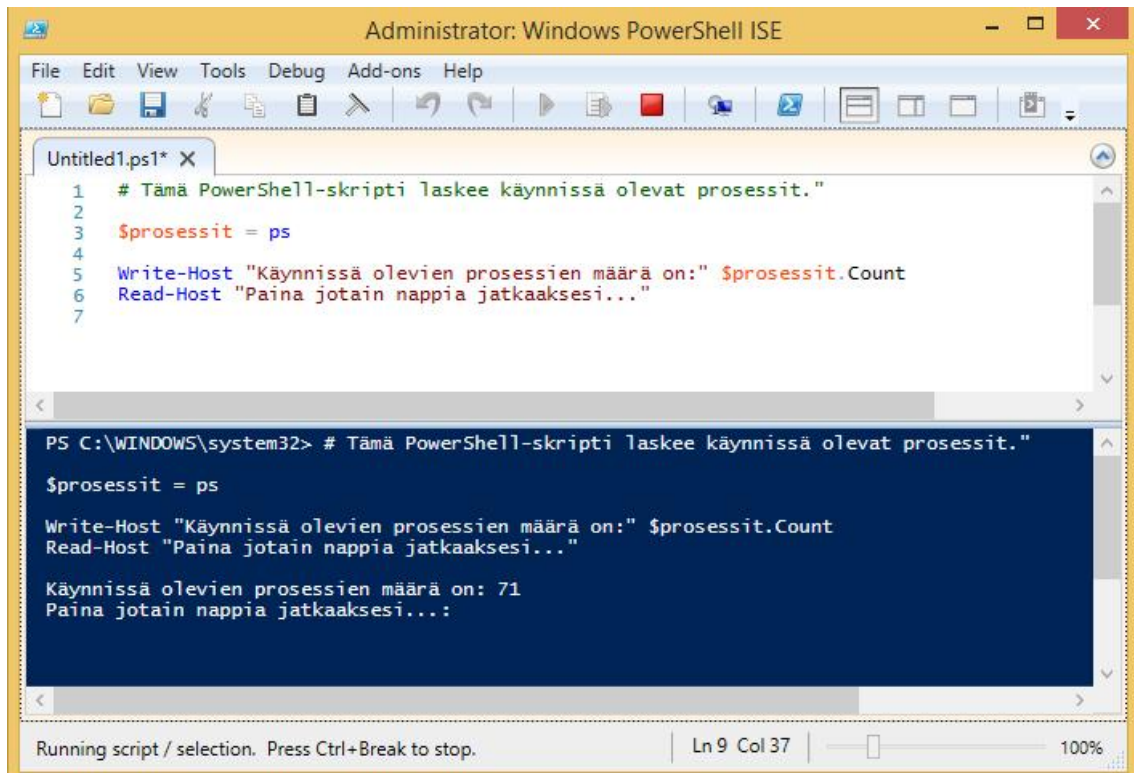
Perinteinen menetelmä soveltuisi helpon käytettävyytensä ja edullisen hintansa puolesta kohderyhmälle myös jatkossa, mutta toimimattomuus Windows 7:n UAC:n kanssa muodostuu ylitsepääsemättömäksi ongelmaksi.



### 3.1.3 Windows PowerShell

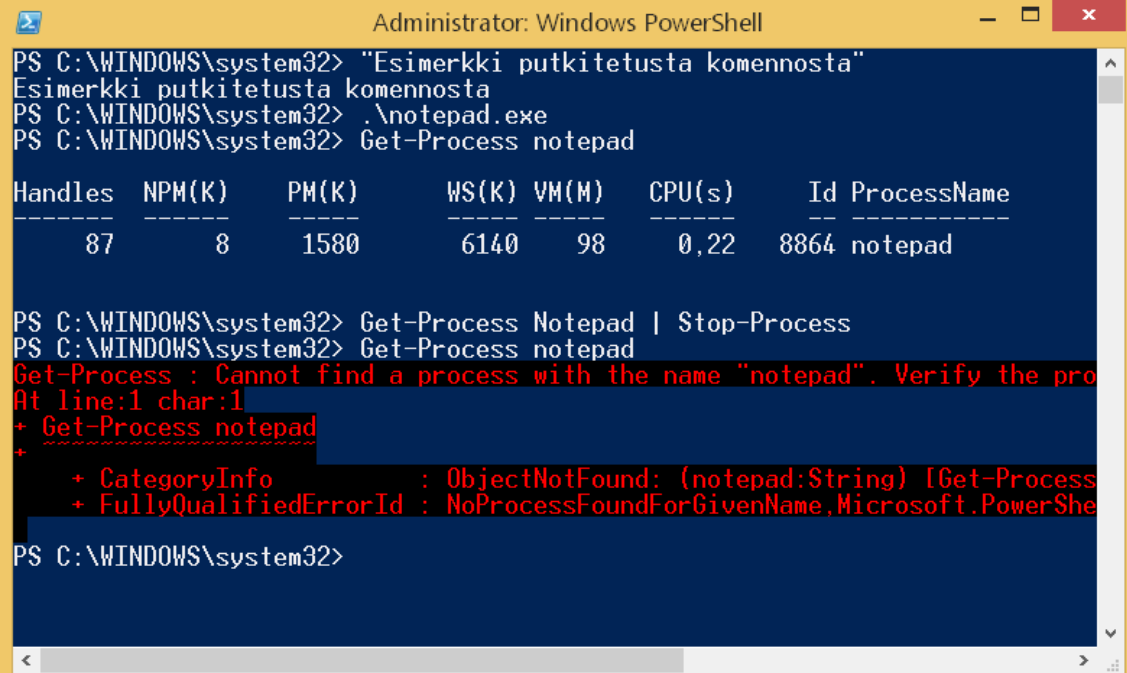
#### Taustaa

PowerShell on IT-ammattilaisten työkalu ympäristöjen ylläpitoon ja automatisointiin. PowerShellia on mahdollista käyttää joko konsolissa tai graafisessa kehitysympäristössä, PowerShell ISE:ssä (kuva 1).



KUVA 1. PowerShell ISE:n käyttöliittymä ja yksikertainen PowerShell-skripti

PowerShellillä voidaan suorittaa yksittäisiä komentoja, useampia putkitettuja komentoja, tai skriptejä. PowerShell-skriptin tunnistaa aina tiedostopäätteestä ps1. (12.) Kuvassa 2 esitellään yksinkertaisesti, kuinka tehokkaasti PowerShellissä voidaan hyödyntää komentojen putkitusta. Get-Process-komennon avulla etsitään käynnissä olevia Notepad-nimisiä prosesseja. Tiedot välitetään putken kautta Stop-Process-komennolle, joka osaa tulkita annetun syötteen ja sulkea löydetty prosessit.



```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> "Esimerkki putkitetusta komennosta"
Esimerkki putkitetusta komennosta
PS C:\WINDOWS\system32> .\notepad.exe
PS C:\WINDOWS\system32> Get-Process notepad

Handles   NPM(K)    PM(K)      WS(K) VM(M)    CPU(s)      Id ProcessName
-----
        87         8       1580       6140     98         0,22    8864 notepad

PS C:\WINDOWS\system32> Get-Process Notepad | Stop-Process
PS C:\WINDOWS\system32> Get-Process notepad
Get-Process : Cannot find a process with the name "notepad". Verify the pro
At line:1 char:1
+ Get-Process notepad
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (notepad:String) [Get-Process]
+ FullyQualifiedErrorId : NoProcessFoundForGivenName,Microsoft.PowerShe
PS C:\WINDOWS\system32>
```

KUVA 2. PowerShell-konsoli ja esimerkki komentojen putkituksesta

Microsoft on sisällyttänyt PowerShellin jokaiseen työasema- ja palvelinkäyttöjärjestelmäänsä Windows 7:stä lähtien. Ohjelma on myös asennettavissa erillisen asennuspaketin avulla Windows XP- ja Windows 2003 Server -käyttöjärjestelmille. PowerShellin uusin versio 4.0 löytyy oletuksena Windows 8.1- ja Windows Server 2012 R2 -käyttöjärjestelmistä.

Microsoftin panostaa tulevaisuudessa yhä enemmän interaktiivisten PowerShell-skriptien käyttöön palvelinten ja ohjelmistojen ylläpidossa. Yhtiö on julkaissut Windows Server 2012 -palvelinkäyttöjärjestelmänsä myös riisuttuna Core-versiona, josta puuttuu graafinen käyttöliittymä. Core-palvelimen ylläpitoa hoidetaan Unix-käyttöjärjestelmän tapaan konsolista PowerShell-komentojen avulla. Unix-ylläpitäjiä on lähestytty lisäämällä mahdollisuus operoida PowerShellissä näennäisesti Unix-komentojen avulla. Esimerkiksi hakemistolistaus saadaan esiin sekä `dir`- että `ls`-komentoilla. Tämä on toteutettu alias-nimien avulla, joten komento `ls` käynnistää oikeasti perinteisen `dir`-ohjelman. Unix-ohjelmia ei siis ole käännetty Windowsille, mutta Unix-ylläpitäjien siirtymi-

sestä Windows-maailmaan on tehty helpompaa. PowerShellistä on tulossa tärkeä työkalu jokaisen IT-ammattilaisen työhön. (10.)

### **Ominaisuudet**

PowerShellin avulla pystytään täydellisesti hallitsemaan kaikkia käyttöjärjestelmän osia ja tietoja niin paikallisesti kuin etänä. PowerShell ei tuo valmista ratkaisua ohjelmistojen jakeluun, mutta tuote mahdollistaa oman jakelumenetelmän kehittämisen tarvittavilla ominaisuuksilla. Jakelumenetelmän toteuttaminen PowerShellin avulla vaatii ohjelmoinnin, PowerShellin ja Windows-käyttöjärjestelmän tuntemusta. Toteuttamalla jakelumenetelmä itse saadaan ohjelmiston toimittajan näkökulmasta mahdollisimman hyvin käyttötarkoitukseen soveltuva ratkaisu.

### **Soveltuvuus**

PowerShellin avulla toteutettu jakelumenetelmä sopii hyvin kohderyhmän käyttöön, sillä kehitystyön hoitaa päivitettävän ohjelmiston toimittaja itse. Ohjelmiston asettamat vaatimukset ja rajoitukset tulee näin huomioitua automaattisesti ratkaisussa. Asiakkaiden toiveita huomioidaan menetelmän kehitystyön alusta saakka siltä osin, kuin niitä on mahdollista toteuttaa. Näin saadaan kehitettyä asiakkaiden tarpeisiin soveltuva kokonaisuus.

Alustojen yhteensopivuus on varmistettu, sillä PowerShell löytyy jokaisesta työasemasta Windows 7 -versiosta lähtien ja on asennettavissa myös Windows XP -käyttöjärjestelmille erillisellä asennuspaketilla. Jakelumenetelmää voidaan päivittää tarpeen vaatiessa, mikäli tulevaisuudessa uudemmilla käyttöjärjestelmän versioilla havaitaan yhteensopivuusongelmia. Microsoft on ilmoittanut, että uusissa PowerShellin versioissa tullaan säilyttämään yhteensopivuus vanhempiin versioihin, mikä on hyvä asia jakelumenetelmän ylläpidollisesta näkökulmasta.

Edellä mainituilla perusteluilla itse kehitettävä PowerShelliin perustuva jakelumenetelmä todettiin parhaaksi ja ainoaksi mahdollisuudeksi toteuttaa vaatimusmäärittelyn mukainen ohjelmiston jakelumenetelmä asiakkaille.

### 3.2 Aktiviteettikaavio

Ratkaisun aktiviteettikaavio on esitetty kuvassa 3. Käyttäjä kirjautuu työasemalle, jolloin työasemaohjelmien päivitys käynnistyy kutsuen SoftwareUpdate.cmd-skriptiä. SoftwareUpdate.cmd tekee työasemalle käyttöjärjestelmän versioon ja asiakkaan ympäristöön liittyviä muutoksia ja käynnistää sitten PowerShell-skriptin InitInstall.ps1. InitInstall.ps1-skripti tarkistaa työasemalle kirjautuneen aktiivisen käyttäjän istunnon ID-numeron ja käynnistää UpdateSoftwareClient.ps1-skriptin tuohon istuntoon. UpdateSoftwareClient.ps1-skripti tarkistaa työasemalta päivitystä tarvitsevat ohjelmat, päivittää ne ja informoi käyttäjää päivityksien etenemisestä.



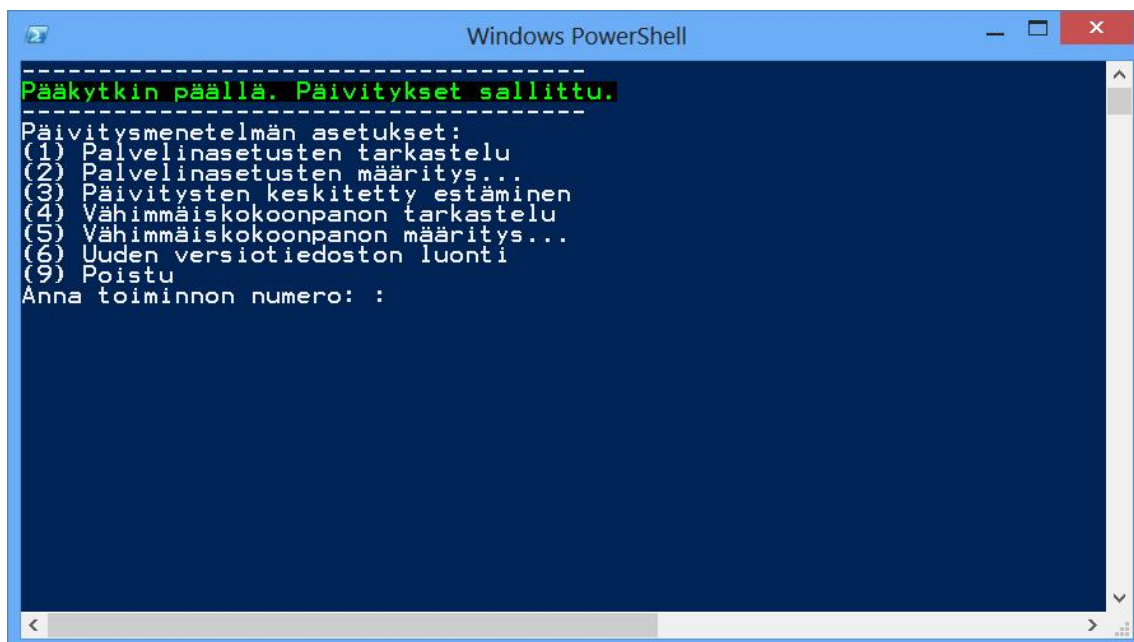
KUVA 3. Aktiviteettikaavio

### 3.3 Käyttöliittymät

Jakelumenetelmän käyttöliittymä suunniteltiin mahdollisimman kevyeksi ja yksinkertaiseksi käyttäjän kannalta. Siinä ei ole graafisia elementtejä, vaan konfigurointi on täysin tekstipohjainen. Käyttöliittymän avulla käyttäjä määrittelee menetelmälle ympäristökohtaiset asetukset ja sallii tai estää jakelun keskitetysti.

#### 3.3.1 Palvelimen käyttöliittymä

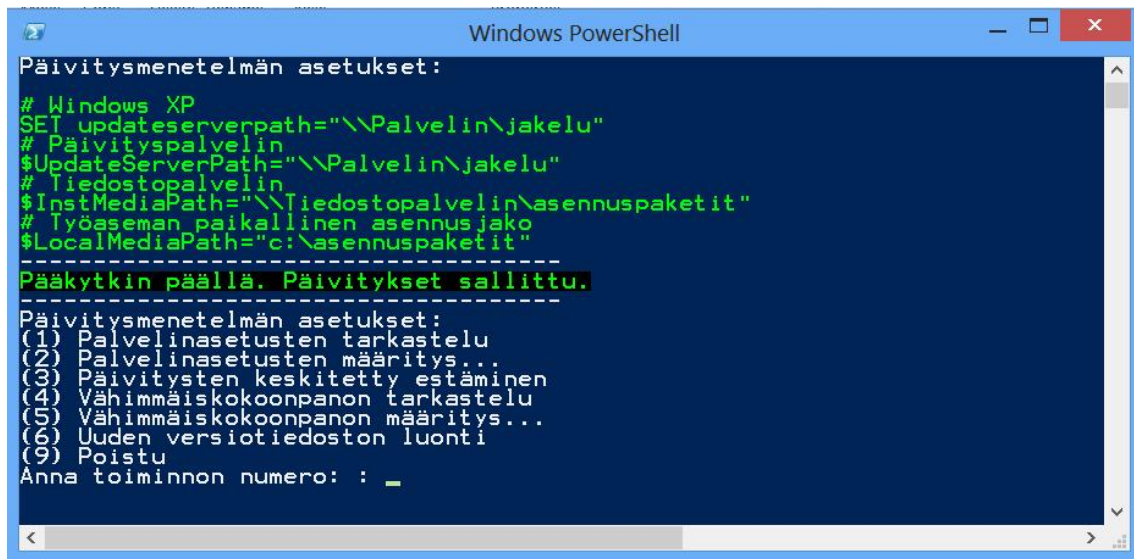
Jakelumenetelmän asetusten hallinta toteutettiin interaktiivisena PowerShell-skriptinä (kuva 4). Ikkunan yläreunassa ilmoitetaan jakelumenetelmän pääkytkimen asento. Käyttäjä liikkuu valikossa numerovalintojen avulla.



KUVA 4. Palvelimen käyttöliittymä jakelumenetelmän konfigurointiin.

#### Palvelinasetusten tarkastelu

Käyttäjä voi tarkistaa voimassa olevat ympäristökohtaiset palvelinmäärittäykset, joita ovat päivityspalvelimen polku, tiedostopalvelimen polku sekä työaseman paikallinen kansio (kuva 5).



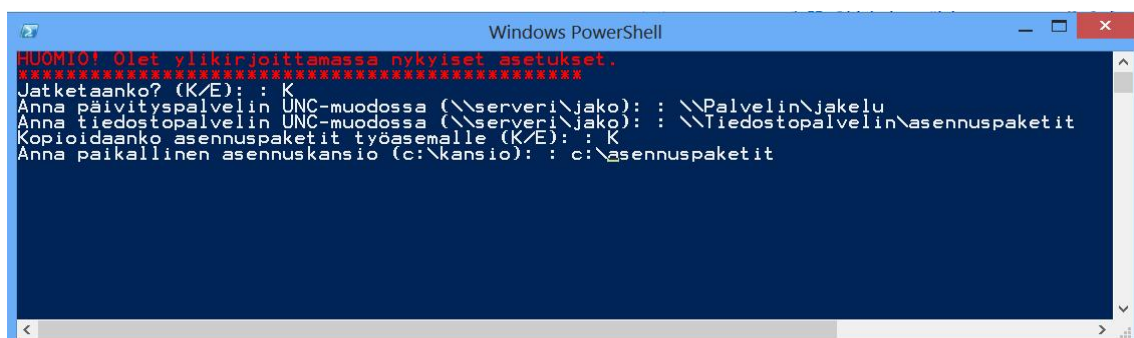
```
Windows PowerShell

Päivitysmenetelmän asetukset:
# Windows XP
SET updateserverpath="//Palvelin\jakelu"
# Päivityspalvelin
$updateServerPath="//Palvelin\jakelu"
# Tiedostopalvelin
$InstMediaPath="//Tiedostopalvelin\asennuspaketit"
# Työaseman paikallinen asennusjako
$LocalMediaPath="c:\asennuspaketit"
-----
Pääkytkin päällä. Päivitykset sallittu.
-----
Päivitysmenetelmän asetukset:
(1) Palvelinasetusten tarkastelu
(2) Palvelinasetusten määrittäminen...
(3) Päivitysten keskitetty estäminen
(4) Vähimmäiskokoonpanon tarkastelu
(5) Vähimmäiskokoonpanon määrittäminen...
(6) Uuden versiotiedoston luonti
(9) Poistu
Anna toiminnon numero: : _
```

KUVA 5. Palvelinasetusten tarkastelu

### Palvelinasetusten määrittäminen

Käyttäjä määrittelee ympäristökohtaiset palvelinasetukset UNC-polun muodossa. Päivityspalvelimeksi annetaan UNC-polku jakelumenetelmän asennuskansiolle, siis sama polku, mistä käyttäjä on käynnistänyt hallintakäyttöliittymän. Tiedostopalvelimeksi määritellään UNC-polku asennuspaketit sisältävään jakoon. Käyttäjä valitsee lopuksi, halutaanko työasemilla käyttää paikallista asennuskansiota tiedostopalvelimen sijasta (kuva 6). Paikallinen asennuskansio määritetään työasemien näkökulmasta absoluuttisena polkuna, esimerkiksi "C:\asennuspaketit".



```
Windows PowerShell

HUOMIO! Olet ylikirjoittamassa nykyiset asetukset.
*****
Jatketaanko? (K/E): : K
Anna päivityspalvelin UNC-muodossa (\\serveri\jako): : \\Palvelin\jakelu
Anna tiedostopalvelin UNC-muodossa (\\serveri\jako): : \\Tiedostopalvelin\asennuspaketit
Kopioidaanko asennuspaketit työasemalle (K/E): : K
Anna paikallinen asennuskansio (c:\kansio): : c:\asennuspaketit
```

KUVA 6. Palvelinasetusten määrittäminen

## Päivitysten keskitetty estäminen tai salliminen

Jakelumenetelmä voidaan millä tahansa ajan hetkellä kytkeä pois päältä ja takaisin päälle tämän valinnan avulla. Käyttöliittymän yläreunassa ilmoitetaan, onko päivityksien jakelu työasemille tällä hetkellä estetty vai sallittu.

## Vähimmäiskokoonpanon tarkastelu

Vähimmäiskokoonpanon tarkastelu -valinta listaa määritetyt työasemaohjelmat, joiden on löydettävä jokaiselta työasemalta. Mikäli jokin listalla ilmoitetuista sovelluksista puuttuu työasemalta, tulee jakelumenetelmä asentamaan sen pakotetusti työasemalle (kuva 7).



```
Windows PowerShell

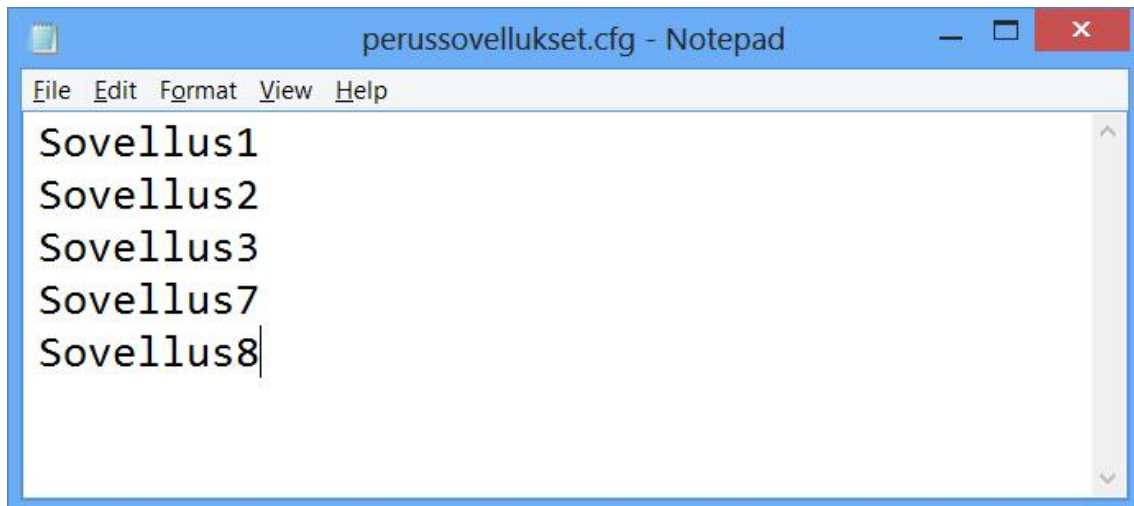
Vähimmäiskokoonpano oltava työasemilla:
Sovellus1
Sovellus2
Sovellus3
Sovellus4
Sovellus5
-----
Pääkytkin on pois päältä! Päivitykset estetty.
-----
Päivitysmenetelmän asetukset:
(1) Palvelinasetusten tarkastelu
(2) Palvelinasetusten määrittäminen...
(3) Päivitysten keskitetty salliminen
(4) Vähimmäiskokoonpanon tarkastelu
(5) Vähimmäiskokoonpanon määrittäminen...
(6) Uuden versiotiedoston luonti
(9) Poistu
Anna toiminnon numero: :
```

KUVA 7. Vähimmäiskokoonpanon tarkastelu

## Vähimmäiskokoonpanon määrittäminen

Vähimmäiskokoonpanon määrittäminen -valinta avaa Notepad-ohjelmalla listauksen, jonne voidaan lisätä tai poistaa työasemaohjelmia vähimmäiskokoonpanoon (kuva 8). Muutoksien jälkeen tallennetaan ja suljetaan tiedosto Notepadissa normaaliin tapaan. Asetusohjelman käyttöliittymä pysyy taustalla käynnissä koko muutoksen ajan.





*KUVA 8. Vähimmäiskokoonpanon määrittäminen*

### **Uuden versiotiedoston luonti**

Käyttäjä luo tällä valinnalla vertailukohtana käytettävän versiolistauksen työasemalla, jolle on esiasennettuna uusimmat työasemaohjelmien versiot.

#### **3.3.2 Työaseman näkymä**

Loppukäyttäjää tiedotetaan päivityksen etenemisestä näytön alakulmassa näytettävän puhekuplan avulla. Kuvassa 9 esitellään käyttäjän näkemä ilmoitus ohjelman asennuksen alkamisesta. Ilmoitus toteutettiin Microsoft Windows.System.Forms -nimiavaruuden NotifyIcon-luokan ShowBalloonTip-metodilla, jolla voidaan toteuttaa graafisia ilmoituksia käyttöjärjestelmän työkaluriville. (13.)



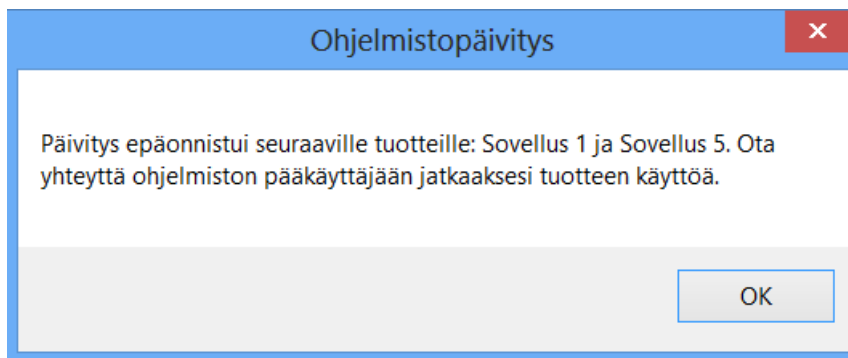
*KUVA 9. Päivityksen aikainen näkymä työasemalla*

Käyttäjät saavat myös ilmoituksen jakelumenetelmän päätettyä ohjelmiston tarkistuksen (kuva 10).



*KUVA 10. Näkymä ohjelmistopäivityksen onnistumisesta*

Epäonnistuneista päivityksistä laaditaan lopuksi yhteenveto, joka näytetään näytön keskelle esiin hyppäävän ikkunan avulla (kuva 11). Samalla ohjeistetaan ottamaan yhteyttä ohjelmiston pääkäyttäjään, mutta tekstiä on mahdollista räätälöidä asiakaskohtaisesti. Ilmoitus toteutettiin Windows Script Host -objektin avulla. (14.)



*KUVA 11. Ilmoitus työasemalla päivityksien epäonnistumisesta*

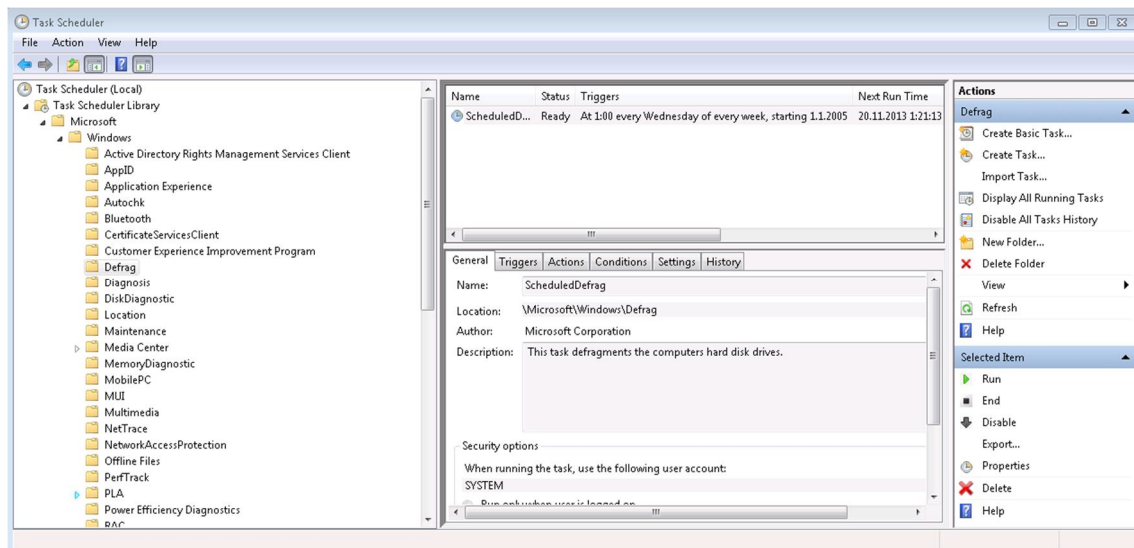
### **3.4 Menetelmän käynnistäminen työasemilla**

Tekninen toteutustapa jakelumenetelmän käyttöönottamiseksi työasemilla jätettiin avoimeksi määrittelyvaiheessa. Asiakkaiden käyttämän ohjelmiston arkkitehtuurin asettama vaatimus on, että työasemaohjelmisto on päivitettävä ennen käyttöä, kun palvelinohjelmisto päivitetään. Koska työasemat ovat usein suljettuina palvelinpäivityksen ajankohtana, tulee työasemaohjelmiston päivityksen onnistua asynkronisesti, käyttäjän työasemalle kirjautumisen yhteydessä.

Asynkronisuutta ei voida toteuttaa pelkästään palvelinpuolen toteutuksena, vaan työasemalla täytyy olla laukaisin, joka käynnistää päivitysmenetelmän työasemalle kirjautumisen yhteydessä. Laukaisin voidaan toteuttaa käyttöjärjestelmän tehtävien ajoittamisen avulla.

### 3.4.1 Tehtävien ajoittaminen

Microsoft hyödyntää Windows 7:ssä ajoitettuja tehtäviä käyttöjärjestelmän säännöllisessä huoltamisessa, kuten levyjen eheyttämisessä ja tietoturvapäivityksissä. Kuvasta 12 on nähtävissä, että ajoitettavia tehtäviä on jo oletuksena käytössä hyvin monessa käyttöjärjestelmän osassa.



KUVA 12. Ajoitettujen tehtävien hallintakonsoli

Koska Microsoft luottaa itsekin ajoitettujen tehtävien hyödyntämiseen käyttöjärjestelmän toiminnan kannalta tärkeissä toiminnoissa, pidettiin laukaisimen toteuttamista ajoitetun tehtävän avulla luotettavana vaihtoehtona. Työasemille päätettiin luoda ajastettu tehtävä, joka aktivoituu käyttäjän kirjautumisen yhteydessä.

### 3.4.2 Ajoitetun tehtävän levittäminen työasemille

Ajoitetun tehtävän luontiin työasemille täytyi suunnitella luotettava ja nopea keino. Sopiva työkalu tähän tarkoitukseen löytyi Active Directoryn ryhmäkäytännöstä eli GPO:sta. GPO:n avulla voidaan luoda ajoitettuja tehtäviä usealle työasemalle yhdellä kertaa. GPO:t päivittyvät toimialueen työasemille joka 90 minuutin välein sekä käyttäjän kirjautuessa työasemalle. GPO varmistaa, että tehtävä luodaan uudelleen, mikäli käyttäjä onnistuu sen vahingossa tai tarkoituksellisesti poistamaan.

GPO on kohdistettava ainoastaan ohjelmiston toimittajan ohjelmistoa käyttäville työasemille. Tätä varten täytyy AD:ssa olla yksi tai useampi organisaatiosyksikkö eli OU, jonne päivitettävät työasemat sijoitetaan ja GPO linkitetään. Päivityksen ulkopuolelle jäävät työasemat sijoitetaan, tai jätetään, omiin OU-haaroihin. Tällä varmistetaan, että jakelumenetelmä ei aktivoidu niillä työasemilla, joissa päivitettävää ohjelmistoa ei ole tarkoitus käyttää.

GPO:n luontiin tarvitaan toimialueen järjestelmänvalvojan (Domain Admin) oikeudet tai erikseen määritetyt käyttöoikeudet toimialuepuun tietyille tasolle. Lisäksi tarvitaan Group Policy Management Console, joka löytyy yleensä AD:n ohjauskoneelta, mutta on myös asennettavissa erilliselle työasemalle tai palvelimelle.

## 4 TOTEUTUS

Projektin määrittely- ja suunnitteluvaiheessa ajateltiin, että jakelumenetelmä olisi toteutettavissa yhden PowerShell-skriptin avulla. Toteutusvaiheessa havaittiin, että määrittelyn mukainen ratkaisu tulisi kuitenkin vaatimaan useamman skriptin käyttämistä. Pystyin uudelleenkäyttämään monia suunnitteluvaiheessa hyväksi havaitsemiani funktioita, mikä nopeutti projektin etenemistä. Toteutin jakelumenetelmää funktio kerrallaan ja testasin uudet ominaisuudet omassa kehitysympäristössäni kahdella eri käyttöjärjestelmällä reaaliajassa. Toteutusvaiheessa havaittiin ongelma ajoitetun tehtävän käyttäytymisessä, sillä asennukset eivät käynnistyneet työasemilla UAC:n estäessä suorituksen.

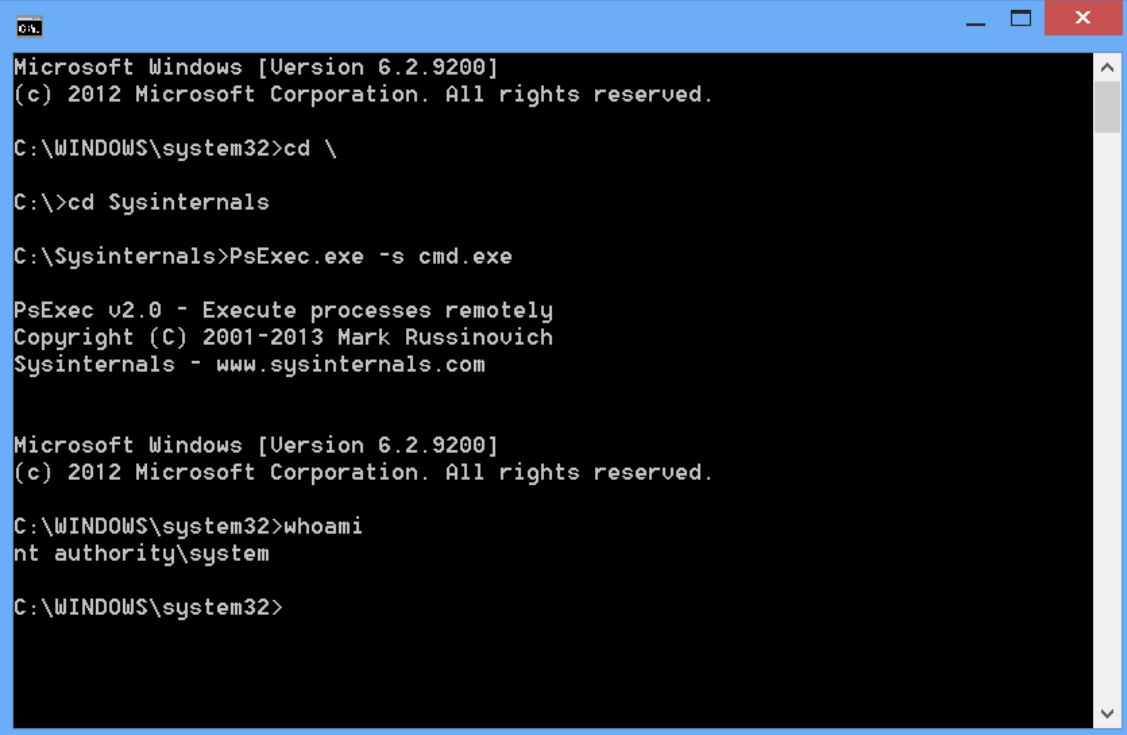
### 4.1 Ratkaistavat ongelmat

#### 4.1.1 Asennuksen vaatimat käyttöjäoikeudet

Ohjelmiston asennuspakettien suorittaminen edellyttää järjestelmänvalvojan oikeuksia työasemalle. Yksi menetelmälle asetettuja vaatimuksia oli suoritua asennuksista loppukäyttäjän käyttäjä-tason oikeuksilla.

UAC tuo mukanaan merkittäviä muutoksia käyttöjäoikeuksien käsittelyyn. Työasemalle kirjautuneella järjestelmänvalvojallakin on tavallisesti samat valtuutukset käyttöjärjestelmään kuin peruskäyttäjän tunnuksella. Kun järjestelmävalvoja yrittää tehdä käyttöjärjestelmän tai kiintolevyn tietoja muuttavan toiminnon, ilmestyy näytölle UAC-dialogi, jossa pyydetään lupaa käyttöoikeuksien nostamiseen tarvittavalle tasolle. Esimerkkinä voidaan käyttää työaseman kellonajan asettamista, mikä vaatii kellonajan muuttamiseen riittävät valtuutukset. Kun kellonaikaa yritetään vaihtaa, UAC-dialogi ilmestyy pyytäen lupaa muutokselle. Dialogin hyväksyminen itseasiassa valtuuttaa hetkellisesti käyttäjän asettamaan työaseman kellonajan uudelleen. UAC-dialogin kuittaaminen ei onnistu millään ohjelmallisella tai automatisoidulla keinolla, mikä estää yksinkertaisimpien haittaohjelmien toiminnan. Kyseessä on siis Microsoftin suunnittelema ominaisuus tietoturvan parantamiseksi.

Yksi harvoista keinoista UAC:n ohittamiseen skriptiin perustuvassa asennusruutiinissa on suorittaa asennukset paikallisella järjestelmättilillä. Microsoftin sivuilta on ladattavissa ilmainen SysInternals-apuohjelmapaketti, joka sisältää pieniä, mutta tehokkaita työkaluja järjestelmien ylläpitoon. Paketista löytyy konsolisovellus nimeltään Psexec, jonka avulla tietokoneella voidaan suorittaa komentoja halutulla käyttäjätunnuksella, kuten järjestelmättilillä (kuva 13). Kirjautuminen järjestelmättilille Psexecin avulla vaatii järjestelmänvalvojan oikeudet tietokoneelle.



```
C:\WINDOWS\system32>cd \
C:\>cd Sysinternals
C:\Sysinternals>PsExec.exe -s cmd.exe

PsExec v2.0 - Execute processes remotely
Copyright (C) 2001-2013 Mark Russinovich
Sysinternals - www.sysinternals.com

Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>whoami
nt authority\system

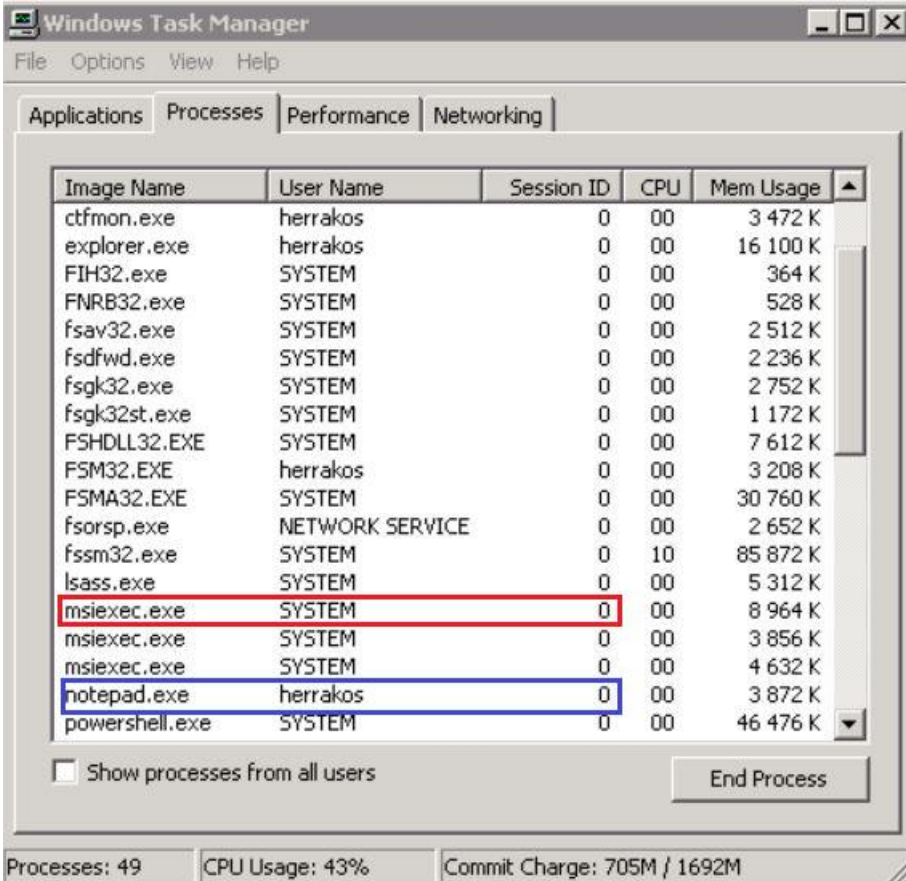
C:\WINDOWS\system32>
```

KUVA 13. Komentokehotteen käynnistäminen järjestelmättilinä

#### 4.1.2 Käyttäjälle näytettävät ilmoitukset

Windows XP:ssä ja sitä vanhemmissa käyttöjärjestelmissä interaktiiviset tapahtumat pystyttiin tarvittaessa näyttämään työasemalle kirjautuneelle käyttäjälle helposti, vaikka interaktiivinen prosessi olisi suoritettu toisena käyttäjänä, kuten järjestelmättilillä. Järjestelmättilin prosessit ovat käynnissä niin kutsutussa konsoli-istunnossa, jonka numero on 0. Myös työasemalle kirjautuneet käyttäjät ohjattiin XP:ssä samaan istuntoon, numero 0. Asia voidaan todeta tarkkailemal-

la prosessien istuntojen numeroita (kuva 14). Sinisellä rajattuna näkyy käyttäjän käynnistämä ohjelma ja punaisella järjestelmätilin suorittama prosessi. Kuten kuvan Session ID -sarakeesta voidaan havaita, molempien tunnuksien prosessit suoritetaan samassa istunnossa numero 0.



| Image Name     | User Name       | Session ID | CPU | Mem Usage |
|----------------|-----------------|------------|-----|-----------|
| ctfmon.exe     | herrakos        | 0          | 00  | 3 472 K   |
| explorer.exe   | herrakos        | 0          | 00  | 16 100 K  |
| FIH32.exe      | SYSTEM          | 0          | 00  | 364 K     |
| FNRB32.exe     | SYSTEM          | 0          | 00  | 528 K     |
| fsav32.exe     | SYSTEM          | 0          | 00  | 2 512 K   |
| fsdfwd.exe     | SYSTEM          | 0          | 00  | 2 236 K   |
| fsgk32.exe     | SYSTEM          | 0          | 00  | 2 752 K   |
| fsgk32st.exe   | SYSTEM          | 0          | 00  | 1 172 K   |
| FSHDL32.EXE    | SYSTEM          | 0          | 00  | 7 612 K   |
| FSM32.EXE      | herrakos        | 0          | 00  | 3 208 K   |
| FSMA32.EXE     | SYSTEM          | 0          | 00  | 30 760 K  |
| fsorsp.exe     | NETWORK SERVICE | 0          | 00  | 2 652 K   |
| fssm32.exe     | SYSTEM          | 0          | 10  | 85 872 K  |
| lsass.exe      | SYSTEM          | 0          | 00  | 5 312 K   |
| msiexec.exe    | SYSTEM          | 0          | 00  | 8 964 K   |
| msiexec.exe    | SYSTEM          | 0          | 00  | 3 856 K   |
| msiexec.exe    | SYSTEM          | 0          | 00  | 4 632 K   |
| notepad.exe    | herrakos        | 0          | 00  | 3 872 K   |
| powershell.exe | SYSTEM          | 0          | 00  | 46 476 K  |

☐ Show processes from all users

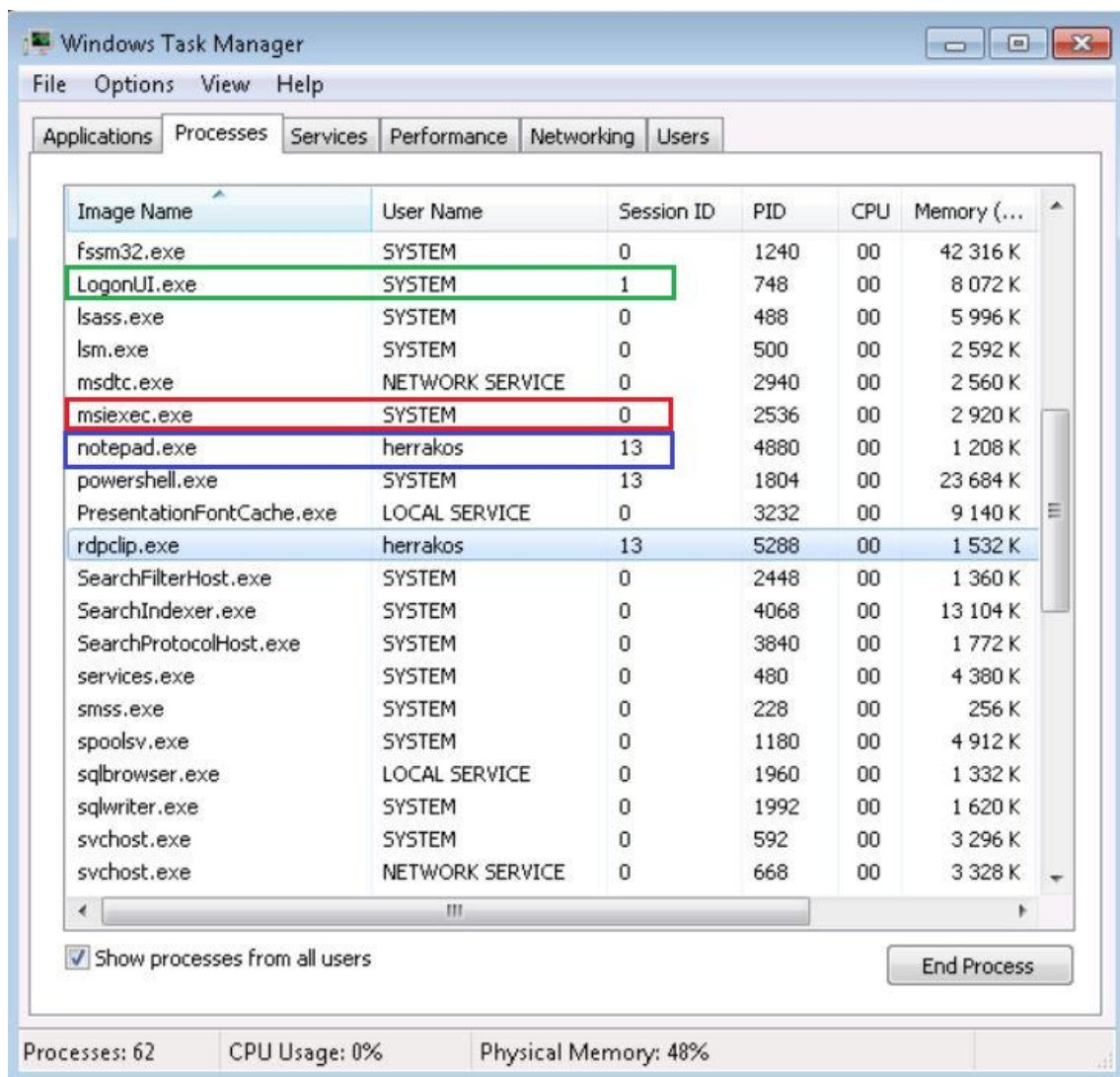
End Process

Processes: 49    CPU Usage: 43%    Commit Charge: 705M / 1692M

KUVA 14. Eri käyttäjien istunnon numerot XP:ssä

Windows Vistassa ja sitä uudemmissa käyttöjärjestelmissä ikkunointi ja istuntojen käsittely muuttui merkittävästi. Paikallisen järjestelmätilin prosessit pyörivät edelleenkin istunnossa numero 0. Käyttöjärjestelmän sisäänkirjautumisdialogi on aina avoinna istunnossa numero 1. Työasemalle kirjautuvalle käyttäjälle annetaan seuraava vapaa istunto. Kuvassa 15 on vihreällä korostettuna kirjautumisdialogi, sinisellä kirjautuneen käyttäjän käynnistämä ohjelma ja punaisella järjestelmätilin prosessi. Käyttäjän työpöydällä ei näy kirjautumisdialogia, eikä

järjestelmätilin suorittamaa asennusohjelmaa, mutta ne ovat silti taustalla käynnissä eri istunnoissa.



KUVA 15. Eri käyttäjien istunnon numerot Windows 7:ssä

Psexecillä suoritettava komento voidaan ohjata haluttuun istuntoon -i <session ID> -valitsimen avulla, missä <session ID> korvataan kirjautuneen käyttäjän istunnon numerolla. Toteutusvaiheessa ohjelman käytössä ilmeni puute, sillä käyttäjän istunnon numero ei ole aina vakio varsinkaan virtuaalisilla työasemilla. Tästä syystä istunnon numeroa ei voitu määrittää valmiiksi skripteihin, vaan asiaan oli suunniteltava jokin dynaaminen ratkaisu.



Ongelma ratkaistiin toteuttamalla erillinen valmisteleva skripti, `InitInstall.ps1`, jonka tehtävä on poimia työasemalle kirjautuneen aktiivisen käyttäjän istunnon numero ja käynnistää `UpdateSoftwareClient.ps1` järjestelmätilin tunnuksella löydettyyn aktiiviseen käyttäjän istuntoon. Tämän avulla päivitykset asennetaan riittävillä käyttöoikeuksilla, mutta tapahtumat näytetään käyttäjän työpöydällä.

#### **4.1.3 Jakelumenetelmän käynnistäminen halutulla hetkellä**

Toteutusvaiheen alussa selvitettiin mahdollisuutta laukaista työasemien ohjelmistojakelu manuaalisesti palvelimelta. PowerShellin `Invoke-Command`-komenton avulla pystytään suorittamaan komentoja joko yhdellä tietokoneella paikallisesti tai usealla etäkoneella samanaikaisesti. (15.)

Toteutusvaiheessa selvitettiin myös mahdollisuutta työasemaohjelmien reaaliaikaiseen päivittämiseen keskitetysti palvelimelta käsin. Tämä tapa toimi hyvin käytännössä pienessä testiympäristössä, mutta työasemien määrän kasvaessa havaittiin vakavana ongelmana se asia, että jokainen työasema ei ollut käynnissä päivityksen hetkellä. Määrittelyvaiheessa tämä asia oli huomioitu, joten synkronisesta päivitystavasta luovuttiin. Reaaliaikaiselle päivitykselle on löytynyt kysyntää, joten asiaa tullaan suunnittelemaan lisää jakelumenetelmän jatkokehitysprojekteissa.

Suunnitteluvaiheessa päätettiin hyödyntää käyttöjärjestelmän ajoitettuja tehtäviä jakelumenetelmän käynnistämisessä. Tehtävälle voidaan antaa liipaisimeksi työasemalle kirjautuminen, minkä avulla päivitykset saadaan käynnistymään silloin, kun loppukäyttäjä kirjautuu aamulla työasemalle. Päivitysmenetelmän skriptit sijoitetaan yhdelle ympäristön palvelimista, jota kutsutaan päivityspalvelimeksi. Päivityspalvelimelle sijoitetusta kansioista luodaan verkkoresurssi, jonne peruskäyttäjille annetaan oikeudet lukea skriptejä sekä kirjoittaa päivityslokia log-kansioon.

Ajoitettu tehtävä luodaan siten, että verkkojaosta määritetään ajettavaksi skripti `SoftwareUpdate.cmd`. Windows mahdollistaa ajoitetun tehtävän suorittamisen toisella käyttäjätunnuksella, joten tehtävän suorittamiseen valitaan paikalliset

järjestelmänvalvojan oikeudet työasemille sisältävä käyttäjätunnus. Tehtävän suoritustasoksi valitaan korotettu, jolloin UAC ei aiheuta ongelmia.

## 4.2 Skriptit

### 4.2.1 Setup.ps1

Jakelumenetelmän konfigurointiin ja hallintaan tarkoitettu interaktiivinen skripti toteutettiin siten, että käyttäjä pystyy sen avulla määrittämään ympäristökohtaiset asetukset ja skripti kirjoittaa ne automattisesti tarvittaviin tiedostoihin.

Seitsemästä päätason valinnasta koostuva valikko toteutettiin switch statement -valintamekanismin avulla. Skripti sisältää myös SaveConfig-funktion asetusten tallettamiseen ja MainSwitch-funktion pääkytkimen asennon tarkistamiseen.

```
# Tämän skriptin avulla hallitaan jakelumenetelmän asetuksia

# Tyhjennetään näyttö
Clear-Host

# Asetusten tallennus
Function SaveConfig()
{...}

# Pääkytkimen tarkistus
function MainSwitch()
{...}

# Interaktiivinen valikko
while(1)
{
    # Get MainSwitch state
    $SwStatus = MainSwitch

    Write-Host '-----'

    if($SwStatus -eq "True")
    {
        Write-Host "Pääkytkin päällä.
        Päivitykset sallittu." -ForegroundColor Green -BackgroundColor Black
    }

    if($SwStatus -eq "False")
    {
        Write-Host "Pääkytkin on pois päältä!
        Päivitykset estetty." -ForegroundColor Red -BackgroundColor White
    }

    Write-Host '-----'
    Write-Host "Päivitysmenetelmän asetukset: "
    Write-Host "(1) Palvelinasetusten tarkastelu"
    Write-Host "(2) Palvelinasetusten määrittäminen..."
    if($SwStatus -eq "True")
```

```

{
    Write-Host "(3) Päivitysten keskitetty estäminen"
}
if($SwStatus -eq "False")
{
    Write-Host "(3) Päivitysten keskitetty salliminen"
}
Write-Host "(4) Vähi mmäi skokoonpanon tarkastelu"
Write-Host "(5) Vähi mmäi skokoonpanon määritys..."
Write-Host "(6) Uuden versiotiedoston luonti"
Write-Host "(9) Poistu"

$choice=Read-Host "Anna toiminnon numero: "

switch($choice)
{
    1 {...}
    2 {...}
    3 {...}
    4 {...}
    5 {...}
    6 {...}
    9 {...}
}
}

```

#### 4.2.2 CreateVersionFile.ps1

Työasemaohjelmien versionumeroiden tarkistaminen toteutettiin PowerShellin Get-ChildItem-komennon avulla. Komennolle annetaan parametrina Windows-rekisterin luettava haara ja sen palauttama tieto tallennetaan muuttujaan. Muuttujan sisällöstä luetaan ohjelmien nimet ja versionumerot moniulotteiseen tauluun. Taulun sisältö kirjoitetaan versiot.cfg-konfiguraatitiedostoon.

```

# Ohjelmiston rekisterihaara 32-bit OS
if([IntPtr]::Size -eq 4)
{
    $regPath = "HKLM:\SOFTWARE\Software"
}

# Ohjelmiston rekisterihaara 64-bit OS
if([IntPtr]::Size -eq 8)
{
    $regPath = "HKLM:\Software\Wow6432Node\Software"
}

Write-Output "Versiotiedot työasemalta $hostname"
" > $UpdateServerPath\cfg\versiot.cfg

function GetSoftwareVersion()
{
    # Ohjelmiston version lukeminen rekisteristä
    $Progs = Get-ChildItem $regPath |
        ForEach-Object {Get-ItemProperty $_.pspath}
}

```

```

for ($i=0; $i -le $Progs.Length - 1; $i++)
{
    $VersionInfo[0] = $Progs[$i] |
        Sort-Object PSChildName |
        foreach {$_.PSChildName}
    $VersionInfo[1] = $Progs[$i] |
        Sort-Object PSChildName |
        foreach {$_.Version}
    $EVersion += $VersionInfo[0] + "|" + $VersionInfo[1]
}
$EVersion >> $UpdateServerPath\cfg\versi ot. cfg
}

```

### 4.2.3 SoftwareUpdate.cmd

SoftwareUpdate.cmd on ajoitetussa tehtävässä kutsuttu komentojonotiedosto, jonka avulla tehdään alustavia asennuksia ja räätälöintejä ennen varsinaisten PowerShell-skriptien suorittamista. Komentojonotiedosto sisältää perusrungon, mikä mahdollistaa eri asioiden suorittamisen käyttöjärjestelmän version perusteella. PowerShell-skriptien ajaminen on oletusarvoisesti estetty käyttöjärjestelmässä set-executionpolicy-määrittelyn avulla. Tämä rajoitus ohitetaan PowerShellin vastaanottamalla parametrilla -executionpolicy bypass. Parametrilla -NoLogo piilotetaan PowerShellin kuvake työkaluriviltä. Parametrin -command jälkeen annetaan suoritettava ps1-tiedosto.

```

: WXP
REM *****
REM                               XP-asennukset/ensi asennukset
REM *****
:: Windows XP -määrittelyt tähän
:: Siirrytään komentojonon asennus-osaan, jos PowerShell on asennettuna
IF EXIST %windir%\system32\WindowsPowerShell\v1.0\powershell.exe GOTO: ASENNUKSET

:: Muuten suoritetaan PowerShellin asennus ensin
:: PowerShellin asennus tälle riville
GOTO: ASENNUKSET

: W7
REM *****
REM                               W7 N7-asennukset/ensi asennukset
REM *****
:: Windows 7 -määrittelyt tähän
::

: ASENNUKSET
:: Käynnistetään PowerShell-skripti Install.ps1

cmd /c powershell.exe -executionpolicy Bypass -NoLogo -command %updateserver-
path\data\Install.ps1

```

#### 4.2.4 InitInstall.ps1

InitInstall.ps1-skripti selvittää työasemalle kirjautuneen aktiivisen käyttäjän istunnon numeron ja käynnistää asennuksen löydettyyn istuntoon paikallisen järjestelmätilin tunnuksella.

Aluksi tarkistetaan käyttöjärjestelmän versio, jonka perusteella skriptissä if-lausekkeiden avulla valitaan suoritettavat komennot. Käyttäjän aktiivisen istunnon numero selvitetään käyttöjärjestelmän eri versioissa hieman eri tavoilla. Seuraavaksi skriptissä muodostetaan komentojono \$cmdline2-muuttujaan, missä yksi annetuista parametreista on aktiivisen istunnon numero. Lopuksi \$cmdline2 välitetään Invoke-Expression-komennolle, joka suorittaa komentojonon paikallisella työasemalla.

```
# Luetaan käyttöjärjestelmän versio muuttujaan
$OSVer = (Get-WmiObject -class Win32_OperatingSystem).Caption

# Windows XP määritykset
if($OSVer -like '*Windows XP*')
{...}

# Määritykset, jos käyttöjärjestelmä ei ole XP
if($OSVer -notlike '*Windows XP*')
{...}

# Komentojonon muodostaminen
$cmdline2 = {...}

# Komentojonon suorittaminen
Invoke-Expression -Command $cmdline2
```

#### 4.2.5 CreateScheduledTask.cmd

Ajoitettu tehtävä voidaan luoda komentoriviltä käyttöjärjestelmän oman schtasks-komennon avulla. Tässä toteutuksessa komennon käyttö on ensisijaisesti suunniteltu käytettäväksi Windows XP -käyttöjärjestelmän kanssa, mutta cmd-tiedostossa on otettu huomioon myös Windows 7.

Käyttöjärjestelmän versio tarkistetaan komentojonon alussa ver-komennon avulla, minkä perusteella suorituksessa hypätään käyttöjärjestelmän versiota vastaavaan funktioon. Ajastettu tehtävä luodaan schtasks-komennolla käyttöjärjestelmän version tunnistamalla syntaksilla.

Schtasks-komennon /create-parametrilla luodaan uusi tehtävä. Windows 7:lle määritetään myös parametri /RL, jolla määritetään tehtävän suoritustaso eli se, suoritetaanko tehtävä rajoitetuilla vai korotetuilla käyttöoikeuksilla. Tällä on suuri merkitys uusissa käyttöjärjestelmissä, sillä järjestelmänvalvojan tunnuksilla ei ole oikeuksia suorittaa asennuksia ennen suoritustason korottamista. Seuraavaksi komennolle annetaan tehtävän suorittamisessa käytettävät tunnukset. Parametrilla /SC määritetään tehtävän suorittamiselle ajankohta, mikä tässä yhteydessä on käyttäjän kirjautuessa. Lopulta ajoitetulle tehtävälle annetaan /TN:llä kuvaava nimi ja /C:llä suoritettava komento.

```
SET updateserverpath="//Palvelin\jaken"

:: Tarkistetaan käyttöjärjestelmän versio
@VER | C:\Windows\System32\FIND "XP" >NUL
IF NOT ERRORLEVEL 1 SET OSVER=WXP
@VER | C:\Windows\System32\FIND "6.1." >NUL
IF NOT ERRORLEVEL 1 SET OSVER=W7

GOTO %OSVER%
GOTO: EOF

: W7
:: Windows 7:n ajastetun tehtävän luonti
%windir%\system32\schtasks /create /RL^
HIGHEST /RU domain\testitep /RP Salasana123^
/SC ONLOGON /TN Update_Software /TR^
"C:\Windows\System32\cmd.exe /c %updateserverpath%\data\SoftwareUpdate.cmd"

GOTO: EOF

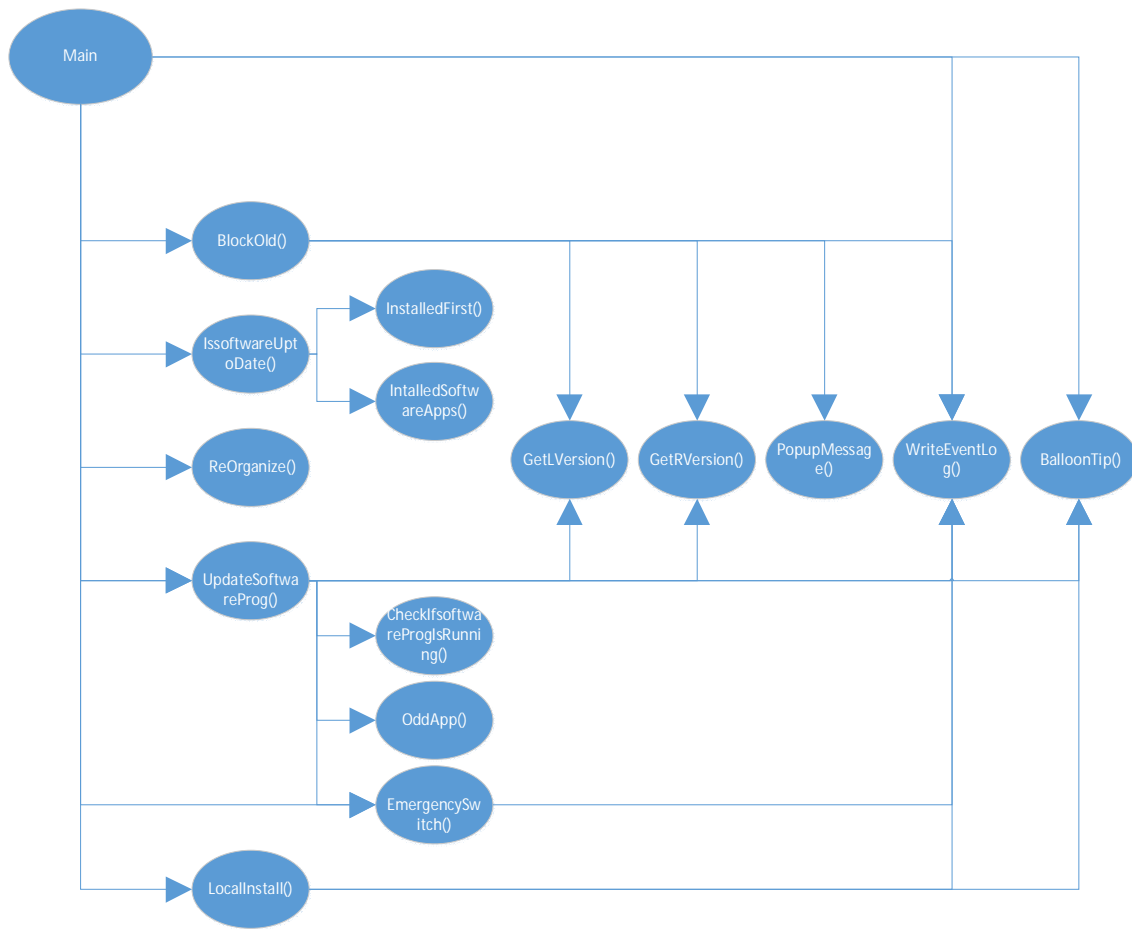
: WXP
:: Windows XP:n ajastetun tehtävän luonti
%windir%\system32\schtasks /create /RU^
domain\testitep /RP Salasana123 /SC^
ONLOGON /TN Update_Software /TR^
"C:\Windows\System32\cmd.exe /c %updateserverpath%\data\SoftwareUpdate.cmd"

GOTO: EOF
```

#### 4.2.6 UpdateSoftwareClient.ps1

##### Kutsupuu

Kuvassa 16 on esitelty UpdateSoftwareClient.ps1-skriptin kutsupuu, josta ilmenevät skriptin sisäisten funktioiden riippuvuudet.



KUVA 16. *UpdateSoftwareClient.ps1* -ohjelman kutsupuu

### LocalInstall-funktio

LocalInstall-funktio tarkistaa ensin, löytyykö työasemalta asennuspaketeille kansio. Kansion löytyessä verrataan sisältöä tiedostopalvelimen kansioon. Mikäli tiedostopalvelimelta löytyy uutta sisältöä tai tiedostojen aikaleimat eivät täsmää, ladataan uusi ja muuttunut sisältö paikalliseen kansioon.

### EmergencySwitch-funktio

EmergencySwitch on pääkytkimen tilaa tarkkaileva funktio, jota kutsutaan useassa skriptin eri vaiheessa. Mikäli funktio havaitsee pääkytkimen tilan olevan pois päältä, päätetään skriptin suorittaminen välittömästi.

### **IssoftwareUptoDate-funktio**

IssoftwareUptoDate-funktio selvittää asennetut työasemaohjelmien versionumerot ja lukee palvelimelta vertailukohdaksi muodostetun versiotiedoston. Funktio vertaa ohjelmien versionumeroita keskenään ja muodostaa tämän perusteella päivitettävien ohjelmien listan, joka palautetaan pääohjelmalle. Listaa tullaan hyödyntämään myöhemmin toisessa funktiossa.

### **ReOrganize-funktio**

Asennusjärjestyksestä huolehtiminen oli yksi tärkeitä vaatimuksia jakelumene-  
telmälle. Pääohjelma kutsuu ReOrganize-funktiota ja antaa sille parametrina päivitettävien ohjelmien listan. Funktio järjestää päivitettävät ohjelmat ennalta määritettyyn järjestykseen ja palauttaa uudelleenjärjestellyn listan pääohjelmalle.

### **CheckIfsoftwareProgsRunning-funktio**

Työasemaohjelmat eivät saa olla käytössä päivityksen alkaessa. CheckIfsoftwareProgsRunning-funktio lukee ohjelmiston asennuskansista kaikkien suoritettavien ohjelmien nimet muuttuinaan. Tämän jälkeen funktio pyytää käyttöjärjestelmää sulkemaan listalta löytyvät ohjelmat. Tätä funktiota kutsutaan useita kertoja skriptin suorituksen aikana.

### **UpdateSoftwareProg -funktio**

Ohjelmien asennuksesta vastaa UpdateSoftwareProg-funktio, jolle pääohjelma välittää parametrina asennusohjeiden asennusjärjestyksestä noudattavan listan päivitettävistä ohjelmista.

Ensin kutsutaan pääkytkimen asennon tarkistavaa EmergencySwitch-funktiota, joka keskeyttää asennuksen tarvittaessa. Tämän jälkeen kutsutaan CheckIfsoftwareProgsRunning-funktiota, joka sulkee käynnissä olevat ohjelmiston prosessit. Tämän jälkeen kutsutaan GetLVersion- ja GetRVersion-funktioita, joille annetaan parametrina päivityksessä olevan sovelluksen nimi. Funktiot palauttavat työasemalle asennetun sovelluksen version ja palvelimen versiotiedostossa määritetyn version.



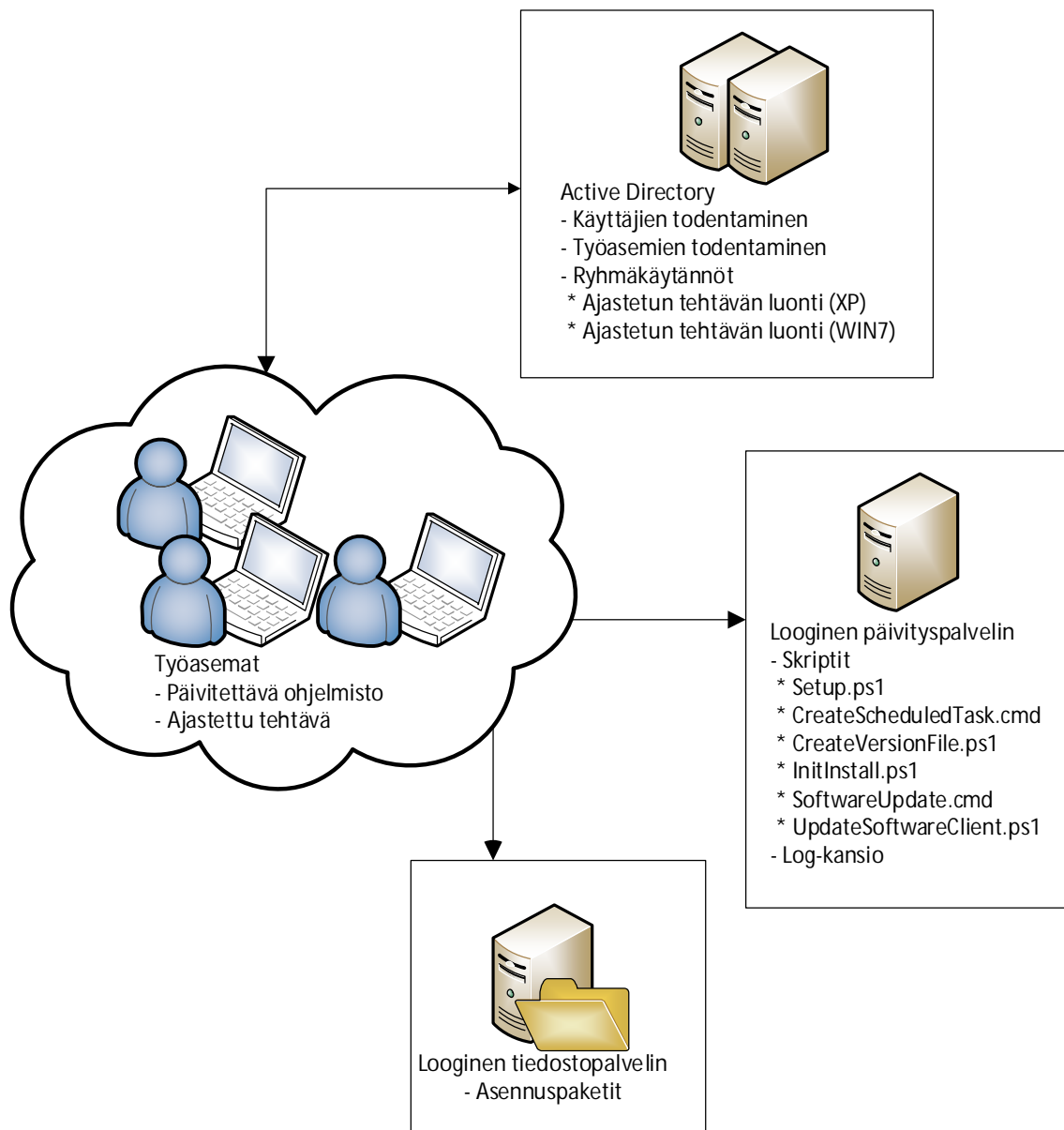
Käyttäjälle näytetään BalloonTip-funktion avulla ilmoitus alkavasta päivityksestä ja kirjoitetaan sama viesti käyttöjärjestelmän lokiin. Tämän jälkeen funktio muodostaa asennuspaketille UNC-polun. Seuraavaksi muodostetaan asennuslauseke, joka sisältää asennuksessa käytettävän komennon, asennuspaketin UNC-polun sekä asennuksessa käytettävät parametrit. Asennuksen jälkeen kirjoitetaan lokimerkinnot käyttöjärjestelmän sovelluslokiin sekä asennuksen lopputuloksen mukaisesti palvelimen log-kansioon työasema.log- tai ERROR-työasema.log-tiedostoon.

Edellä kuvatut rutiinit toistetaan jokaiselle päivitettävien listalta löytyvälle sovellukselle.

### **4.3 Palvelin- ja työasema-arkkitehtuuri**

Kuvassa 17 on esitetty jakelumenetelmän sijoittelukaavio. Työasema hakee GPO:n AD-toimialueelta. GPO:n avulla luodaan työasemalle päivitysmenetelmän käynnistävä ajoitettu tehtävä. Tehtävä aktivoituu loppukäyttäjän kirjautuessa työasemalle.

Ajoitettu tehtävä käynnistää päivityspalvelimen verkkojaosta SoftwareUpdate.cmd -skriptin, mikä aloittaa koko päivitysketjun. Asetuksien mukaisesti päivitysmenetelmä asentaa ohjelmistoon kohdistuvia päivityksiä joko suoraan tiedostopalvelimelta tai työaseman paikallisesta kansioista. Työasema kirjoittaa päivityksen tapahtumista lokia päivityspalvelimen log-kansioon.



KUVA 17. Sijoittelukaavio

## 5 TESTAUS

### 5.1 Pilotointi

Jakelumenetelmälle järjestettiin pilotti kahden eri asiakkaan testiympäristöissä, joista toinen oli keskisuuri ja toinen pieni asiakasorganisaatio. Asennus ja testaus toteutettiin yhteistyössä asiakkaan IT-ylläpidon ja ohjelmiston pääkäyttäjien kanssa. Testissä oli mukana sekä XP- että Windows 7 -työasemia, joille oli tarkoituksena jaella ohjelmiston uusinta versiota.

Ensimmäinen pilotti tehtiin keskisuurella asiakkaalla virtuaalisilla työasemilla. Menetelmän käyttöönotto sujui suunnitelmien mukaisesti. Päivitysvaiheessa havaittiin kuitenkin ongelmia tapahtumien näyttämässä käyttäjälle. Ongelman selvityksen jälkeen syyksi paljastuivat eroavaisuudet oman kehitysympäristön ja asiakkaan testiympäristön käyttöjärjestelmien kieliasetuksissa. Aktiivisen istunnon poiminnassa etsitty sana oli käyttöjärjestelmän lokalisoima, minkä seurauksena istunnon numeroa ei pystytty löytämään ja kaikki tapahtumat ohjautuivat järjestelmätiliin istuntoon. Istunnon numeron hakemisesta vastaavaa InitInstall.ps1-skriptiä muutettiin siten, että sekä englannin- että suomenkieliset käyttöjärjestelmät huomioidaan. Asiakas oli kiinnostunut uuden menetelmän mahdollisuuksista ja siltä saatiin hyviä ajatuksia paranneltavista asioista.

Toinen pilotti järjestettiin pienen asiakkaan uusilla fyysisillä työasemilla. Menetelmän käyttöönotto vaihe sujui hyvin, mutta päivitykset eivät kuitenkaan käynnistyneet asiakkaan työasemilla, eikä virhelokia muodostunut. Ongelman syyksi paljastuivat asiakkaan työasemien poikkeavat ympäristömuuttujat, joista puuttuivat käyttöjärjestelmän omat kansiot. Skripteissä kutsutaan useassa kohdassa käyttöjärjestelmän omia komentoja, joita ei ongelman vuoksi tunnistettu. Vastaavan varalle kaikki skriptien komennot muutettiin käyttämään absoluuttisia polkuja. Muutoksen jälkeen asiakaspilotti saatiin vietyä onnistuneesti päätökseen.

Asiakaspiloteissa tehtyjen havaintojen ja niissä kerättyjen kehitysehdotusten pohjalta tehtiin vielä pienempiä muutoksia ratkaisuun.

## **5.2 Sisäinen testaus**

Ratkaisu asennettiin sisäiseen validointiympäristöön ja kohdistettiin aluksi pienen osaan testiympäristön virtuaalisista Windows 7 -työasemista. Työasemat sijoitettiin AD:ssa omaan OU-haaraan, johon linkitettiin GPO ajoitetun tehtävän luontia varten.

Asiantuntijoiden kanssa järjestettiin testausistuntoja, joissa varmistettiin ensin tehtävien löytyminen työasemilta. Tehtävien luonnissa GPO:n avulla ei havaittu ongelmia. Testauksessa simuloitiin erilaisia työasemilla tapahtuvia muutoksia ja tutkittiin näiden vaikutuksia jakelumenetelmän toimintaan. Testeissä pystyttiin löytämään pieniä korjattavia puutteita, mutta ei merkittäviä ongelmia.

Jakelumenetelmän käyttöä laajennettiin kattamaan sisäisen testiympäristön useimmat työasemat sekä Citrix-farmin palvelimet. Vaikka kehitystyötä ei varsinaisesti tehty Citrix-palvelimia silmällä pitäen, onnistui työasemaohjelmien päivittäminen palvelimilla hyvin. Citrix-palvelimilla päivitystapahtumaa ei haluttu laukaista käyttäjien kirjautumisen yhteydessä, joten ajoitettu tehtävä muutettiin aktivoitumaan kerran vuorokaudessa. Kaiken kaikkiaan jakelumenetelmä suoriutui työasemien päivityksestä erilaisissa tilanteissa hyvin ja sai näin ollen hyväksynnän sisäisessä validoinnissa.

## **5.3 Asiakastestaus**

Seuraavaksi kirjoitettiin jakelumenetelmän käyttöopas ja järjestettiin omille asiantuntijoille koulutuksia. Käyttöoppaaseen sisällytettiin jakelumenetelmän käyttöönoton eri vaiheet, ylläpidollisia asioita sekä yleisiä käytännön esimerkkejä. Käyttöoppaan valmistumisen jälkeen jakelumenetelmää alettiin tarjoamaan asiakaskuntaan. Menetelmän asennuksesta sovittiin heti usean kiinnostuneen asiakkaan kanssa. Asiakkailla oli käytettävissä erilliset testiympäristöt, jotka sopivat hyvin menetelmän testaamiseen.

Testiympäristöissä menetelmän käyttöönotot tehtiin yhteistyössä asiakkaan IT-ylläpitäjien ja ohjelmiston pääkäyttäjien kanssa. IT-ylläpitäjät vastasivat GPO:n luonnista toimitettujen ohjeiden mukaisesti. Ohjelmiston pääkäyttäjiltä saatiin tietoja ympäristökohtaisista vaatimuksista ja he pystyivät myös testaamaan toteutusta työasemilla. Pääkäyttäjät asensivat myös testikoneen, josta luotiin työasemaohjelmien versiot sisältävä vertailutiedosto. Testauksen ohessa asiakkaan ylläpitohenkilöstöä koulutettiin ratkaisun käyttämiseen.

Asiakastestit sujuivat pääpiirteittäin hyvin, eikä itse toteutuksesta löytynyt tässä vaiheessa enää korjattavia virheitä. Testeissä ilmeni silti ajoittain haasteita asiakasympäristöjen vaihtelevuuden takia. Ongelmat saatiin kuitenkin selvitettyä tapauskohtaisesti ja asiakaspalautteet olivat hyviä.

## 6 JATKOKEHITYS

Jakelumenetelmää kehitetään jatkossa asiakasvaatimusten, alustojen tai ohjelmiston muuttuessa sekä mahdollisten vaarantavien virheiden ilmetessä. Yhteensopivuustestaus Windows 8 -käyttöjärjestelmän kanssa toteutetaan sen jälkeen, kun Windows XP on poistunut asiakasympäristöistä ja Windows 8:n käyttöönotot tulevat ajankohtaiseksi. Windows 8 ei todennäköisesti vaadi muutoksia ratkaisuun, sillä Microsoft on luvannut säilyttää PowerShellin yhteensopivuuden vanhempiin versioihin.

### 6.1 Omia havaintoja

Asiakkaiden käyttöönottoprojektien yhteydessä tein muutamia havaintoja tarpeellisista muutoksista toteutuksessa. Nämä muutokset tullaan toteuttamaan ratkaisun jatkokehitysprojeekteissa.

Yksi hyödyllinen ominaisuus olisi saada automaattinen koostettu raportti jakelumenetelmän onnistumisesta työasemilla. Nykyisessä versiossa päivityksen etenemistä on mahdollista seurata työasemien kirjoittamien lokimerkintöjen perusteella. Tämä ei kuitenkaan paljasta, jos päivitysmenetelmä ei käynnisty lainkaan työasemalla. Uusi ominaisuus olisi mahdollista toteuttaa listaamalla AD:n OU-haaran konetilit ja vertaamalla niitä lokimerkintöjä tehneisiin työasemiin. Tämän ominaisuuden avulla asiakkaan ylläpitäjät pystyisivät ennakoivasti tutkimaan ongelmakoneet jo käyttökatkojen aikana tai ainakin nopeasti käyttökatkon päättymisen jälkeen.

Toinen parannettava ominaisuus liittyy lokitiedostojen muodostamiseen. Käytäntö on osoittanut, että joissakin tapauksissa lokitiedostot saattavat olla vaikeita tulkita. Tällä hetkellä on mahdollista, että työasema kirjoittaa lokimerkinnän ensin onnistuneesta ohjelman asennuksesta, mutta lisäksi kirjoittaa samalle ohjelmalle virhemerkinnän lokitiedostoon. Tämä aiheuttaa toisinaan epäselvyyttä lokimerkintöjen tulkitsemisessa. Asian korjaaminen vaatii suurehkoja muutoksia UpdateSoftwareClient.ps1 -skriptin tärkeimmissä funktioissa.

Jakelumenetelmän asennusvaiheessa on tullut usein vastaan käyttöoikeusongelmia konfiguraatitiedostojen muokkaamisessa. Syynä on ollut aina inhimillinen virhe asennusvaiheessa. Tiedosto-oikeuksien tarkistaminen esimerkiksi Setup.ps1 -skriptin avulla vähentäisi näiden virheiden mahdollisuutta.

## **6.2 Asiakaspalautteita**

Ohjelmiston toimittajan ohjelmistolla on tiettyjä vaatimuksia työasemien varusohjelmille. Usealta asiakkaalta on tullut pyyntö, että menetelmä tarkkailisi myös varusohjelmien asennuksia ja päivittäisi ne tarpeen vaatiessa. Asiakkaiden viikailmoitukset liittyvät usein puutteellisiin varusohjelmiin tai niiden väärin versioihin. Mikäli varusohjelmien asennukset pystyttäisiin myös automatisoimaan, vähentäisi se merkittävästi ongelmien lukumäärää.

Asiakkaat ovat myös tiedustelleet mahdollisuudesta käynnistää työasemien päivitykset reaaliaikaisesti kesken työpäivän siten, että loppukäyttäjän ei tarvitse kirjautua uudelleen työasemalle. Tämä ominaisuus on teknisesti toteutettavissa pienin muutoksin nykyiseen ratkaisuun. Asia vaatii kuitenkin huolellista suunnittelua, sillä asiaan liittyy esimerkiksi loppukäyttäjien informoinnista huolehtiminen. Muilta osin asiakkailta saadut palautteet ovat olleet pelkästään positiivisia.

## 7 YHTEENVETO

Opinnäytetyössä määriteltiin, suunniteltiin, toteutettiin ja testattiin ohjelmiston jakelumenetelmä Windows 7 -käyttöjärjestelmälle. Opinnäytetyöni aihe syntyi ohjelmistotoimittajan asiakkaiden tarpeesta löytää korvaava ratkaisu perinteiselle jakelumenetelmälle.

Projekti alkoi vaatimusmäärittelyllä, jossa selvitettiin päivitettävään ohjelmistoon liittyvät vaatimukset ja rajoitukset. Lisäksi kartoitettiin asiakkaiden ja tekniikan vaatimukset. Määrittelyn jälkeen aloitettiin suunnitteluvaihe, jossa suunniteltiin käyttöliittymä, jakelumenetelmän perusrakenne sekä jakelumenetelmän käynnistystapa työasemilla. Toteutusvaiheessa ratkaistiin vielä avoimia ongelmia ja kirjoitettiin skriptit suunnitelmien mukaisesti.

Testausvaiheessa ratkaisua testattiin sisäisissä testiympäristöissä ja järjestettiin pilotteja asiakasympäristöissä. Sisäisessä validoinnissa saadun hyväksynnän jälkeen ratkaisulle laadittiin käyttöopas ja aloitettiin testaukset asiakkaiden testiympäristöissä. Asiakastestit sujuivat pääsääntöisesti hyvin ja asiakkailta saadut palautteet olivat hyviä.

Projektista opin sen, että määrittelyvaiheen tärkeyttä ei voi koskaan korostaa tarpeeksi. Avoimeksi jääneet asiat ja joiltain osin puutteellinen asiakasvaatimuksien kartoitus teettivät projektin edetessä ylimääräistä työtä. Pilottien järjestäminen asiakkaiden kanssa osoittautui hyväksi ratkaisuksi, sillä niiden avulla löydettiin ongelmia, joita omissa testiympäristöissä ei havaittu. Tämän lisäksi saatiin toteutuksesta välitöntä asiakaspalautetta, jonka avulla pystyttiin vielä tekemään pieniä parannuksia lopulliseen toteutukseen.

Jakelumenetelmän jatkokehitysprojekteissa tullaan korjaamaan ensimmäisessä versiossa havaittuja puutteita ja lisäämään uusia ominaisuuksia. Windows 8 -käyttöjärjestelmän yhteensopivuus varmistetaan, kun käyttöjärjestelmän käytönnotot tulevat ajankohtaiseksi asiakasorganisaatioissa.



PowerShell on kaiken kaikkiaan tehokas IT-ylläpidon työkalu ja sillä on tulevaisuudessa merkittävä rooli ympäristöjen hallinnassa. PowerShellin tehokas hyödyntäminen vaatii ohjelmointikielen perusteiden osaamista ja kokemusta Windows-ympäristöjen ylläpidosta. Skriptien laatimisessa yleisestä ohjelmointikokemuksesta on paljon etua, mutta julkista ohjemateriaalia on hyvin saatavilla.

## LÄHTEET

1. Active Directory. 2002. Wikipedia. Saatavissa: [http://en.wikipedia.org/wiki/Active\\_Directory](http://en.wikipedia.org/wiki/Active_Directory). Hakupäivä 17.12.2013.
2. Group Policy. Microsoft. Saatavissa: <http://technet.microsoft.com/en-us/windowsserver/bb310732.aspx>. Hakupäivä 17.12.2013.
3. Organizational Units. 2014. Microsoft. Saatavissa: <http://technet.microsoft.com/en-us/library/cc978003.aspx>. Hakupäivä 29.1.2014
4. PowerShell. 2012. Microsoft. Saatavissa: <http://technet.microsoft.com/en-us/library/ff950685.aspx>. Hakupäivä 17.12.2013.
5. Russinovich, Mark 2013. Sysinternals Suite. Microsoft Technet. Saatavissa: <http://technet.microsoft.com/en-us/sysinternals/bb842062.aspx>. Hakupäivä 17.12.2013.
6. System Center 2012 R2 Configuration Manager. 2014. Microsoft. Saatavissa: <http://www.microsoft.com/en-us/server-cloud/products/system-center-2012-r2-configuration-manager/default.aspx#fbid=cb5IGWDsqhA>. Hakupäivä 29.1.2014.
7. User Account Control. 2013. Wikipedia. [http://en.wikipedia.org/wiki/User\\_Account\\_Control](http://en.wikipedia.org/wiki/User_Account_Control). Hakupäivä 17.12.2013.
8. Documentation Library for System Center 2012 Configuration Manager. 2013. Microsoft. Saatavissa: <http://technet.microsoft.com/en-us/library/gg682144.aspx>. Hakupäivä 11.2.2014.
9. Site Administration for System Center 2012 Configuration Manager. 2013. Microsoft. Saatavissa: <http://technet.microsoft.com/en-us/library/gg682075.aspx>. Hakupäivä 11.2.2014.

10. Managing a Server Core Server. 2012. Microsoft. Saatavissa: <http://technet.microsoft.com/en-us/library/jj574205.aspx>. Hakupäivä 12.11.2013.
11. Understanding and Configuring User Account Control in Windows Vista. 2009. Microsoft. Saatavissa: [http://technet.microsoft.com/en-us/library/cc709628\(v=ws.10\).aspx](http://technet.microsoft.com/en-us/library/cc709628(v=ws.10).aspx). Hakupäivä 12.11.2013.
12. Windows PowerShell. 2013. Wikipedia. Saatavissa: [http://en.wikipedia.org/wiki/Windows\\_PowerShell](http://en.wikipedia.org/wiki/Windows_PowerShell). Hakupäivä 12.11.2013.
13. NotifyIcon Class. 2013. Microsoft. Saatavissa: [http://msdn.microsoft.com/en-us/library/system.windows.forms.notifyicon\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/system.windows.forms.notifyicon(v=vs.110).aspx). Hakupäivä 17.11.2013.
14. Converting the Windows Script Host Popup Method. 2013. Microsoft. Saatavissa: <http://technet.microsoft.com/en-us/library/ff730993.aspx>. Hakupäivä 17.11.2013.
15. Invoke-Command. 2011. Microsoft. Saatavissa: <http://technet.microsoft.com/en-us/library/dd347578.aspx>. Hakupäivä 28.11.2013.